FREE ENTRY COMPETITION
SOFTWARE TO BE WON

DEC 82/JAN 83

75

ZX COMPUTING

MACHINE CODE Pt 2 27

Britain's Biggest Magazine For The Sinclair User

Over 120 pages of information and programs for the ZX Spectrum, ZX80 and ZX81 Computers



PLUS

- 3-D Graphics on Your ZX-81
- Maze-Chaser on the Spectrum
- Education and Business on the Sinclair Computers
- Defeat the Wall With Only a ZX-81!

ZX COM

ZX Computing
Vol. One
Number Three
Oct/Nov 1982

Editor: Tim Hartnell
Editorial Assistant: Helen Bruff
Advertising Manager: Neil Johnstone.
Managing Editor: Ron Harris
Managing Director: T J Connell

Origination and design by MM Design & Print,
145 Charing Cross Road, London WC2H 0EE.

Published by Argus Specialist Publications Ltd,
145 Charing Cross Road, London WC2H 0EE.

CONTENTS

MPUTING

ZX Computing is published bi-monthly on the fourth Friday of the month. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. 01-247 8233. Printed by: Henry Garnett Ltd., Rotherham.

NEW! ZX SPECTRUM TAPE NOW READY!
NEW! EXPANDED DISC VERSIONS FOR
APPLE, PET AND SHARP!

NOW AVAILABLE FOR THE ZX SPECTRUM!

THE VALLEY



© ASP LTD 1982

What are you... Barbarian or Wizard?

Choose your character type carefully... Barbarians recover quickly but their magic doesn't come easily. A Wizard? Slow on the draw and slow to mature...but live long enough and grow wise enough and your lightning bolts are almost unstoppable...

The Valley is a real-time game of adventure and survival. You may choose one of five character types to be your personal 'extension of self' to battle and pit your wits against a number of monsters. Find treasure, fight a Thunder-Lizard in the arid deserts of the Valley, conquer a Kraken in the lakes surrounding the dread Temples of Y'Nagioth or cauterise a Wraith in the Black Tower. In fact, live out the fantasies you've only dared dream about. BUT BEWARE...more die than live to tell the tale.

The Valley runs in a 48K Spectrum (a 16K version will be available shortly) and makes full usage of the excellent colour and graphics capability of the machine. As you battle your way up the ratings your character can be saved onto tape, to re-enter the Valley another day — if you dare!

Full instructions are included with the game, but if you want more detail on the program, a 16 page reprint of the orignal 'Computing Today' article is available at £1.95 all inc.

If you have the courage, the Valley costs only £11.45 all inc. from ASP SOFTWARE, 145 Charing Cross Road, London WC2 0EE.

Please send me...tape(s) of The Valley at £11.45 per tape all inc.
and..........listings at £1.95 each.
I enclose my cheque/Postal Order/International Money Order for: (delete as necessary)
£.....(Made payable to ASP Ltd)
or Debit my Access/Barclaycard
(delete as necessary)

Please use BLOCK CAPITALS
NAME(Mr/Mrs/Miss) ....
ADDRESS ....
.... POSTCODE ....
Signature .... Date ....

ZX/1

ASP SOFTWARE

The times they are a-changin



We've seen a number of new computers, in the under-£150 price range, released in the past few months. Several Japanese machines, including a low-price machine which appears to be an Apple in all but appearance and price, are creating immense interest in the marketplace, and disquiet among homegrown computer manufacturers.

Commodore have a £99.95 colour machine on the way, the £49.95 Binatone computer seems to be a genuine project which will spring fully formed into Woolies stores shortly, and the Jupiter Ace, the Forth-speaking machine from the two key men in the project to build the Spectrum, is building up a sizeable following.

But despite this competition, and despite the inevitable fact that the computer marketplace is fragmenting, the Sinclair computers seem to have a hold on the imagination which no other machine has acquired. Despite immense delivery delays in the past few months, orders for the Spectrum continue to flood into Camberley. Clive Sinclair says that sales of the ZX81 jumped by 500 per cent when he dropped the price from £69.95 to £49.95.

So, whether you decide to buy a ZX81 off the shelf right now, or stand patiently in line for a Spectrum, you'll be in good company. The large number of machines on the market, a strong national users club and many local clubs, and a large body of good software, means the country seems almost purpose-built to ensure that buyers of ZX micros will be helped from every direction to make the most of their machines.

ZX Computing is, of course, part of the life support system which has evolved around the Sinclair computers and in this issue we have much to interest you, no matter which Sinclair machine you have.

The 'large body of good software' I mentioned earlier brings its own problems. It is difficult to know which program to buy, and in this issue reviewer Phil Garratt casts his professional eye over a number of Spectrum packages, including programs with the intriguing titles of Meteor Storm, Space Intruders and Great Britain Limited.

Nick Pearce continues to delve into the ZX81 software explosion, and in this issue looks at Byteman, Namtir Raiders and Space Mission, among others. You'll find, as well, a review of a program which already seems to have attained the status of a ZX81 'classic', the unique MAZOGS from Bug-Byte. We look at hardware add-ons, as well, with an interface to allow your ZX81 to be connected to an ordinary printer being highlighted.

In the last issue, we started several topics which are continued and expanded in this issue. Toni Baker's series on mastering machine code on the Spectrum gets into the complex world of registers and addresses, our BASIC LIFE program of the last issue has become a four-page machine code program, and I expand the material given on teaching your computer to play board games with a couple of programs with full-screen displays.

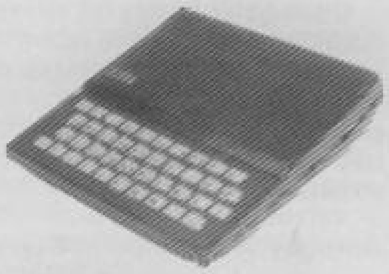
Bubble sorting is discussed along with the use of the Spectrum's ATTR function.

There's much more, as you'll see when you leaf through the pages of this issue, so I suggest you start the serious business right now of making the most of your micro with ZX Computing.

TIM HARTNELL

(Tim Hartnell, the editor of ZX Computing, is author of a number of books, and is co-ordinator of the National ZX Users' Club, 44-46 Earls Court Road, London W8 6EJ, which acts as an umbrella club to local clubs.)

Editor Tim Hartnell looks at the changes occuring in the low price computer market, and previews the contents of this bumper issue of ZX Computing.



Musing on memory

Dear ZX Computing,
At MEMOTECH we were naturally very pleased by the nice things Tim Langdell had to say about us in "RAM, SWEET RAM" — the review of memory packs in your Aug/Sep issue. It has taken 6 months — a long time in computing — to get the review we feel we rightly deserve! But it raised a number of points in which your readers may be interested.

1) Prices.
Our 16K pack now sells at £29.90 including VAT — 5p less than Sinclair! At this price we only offer £10 trade-in for a 64K so the discounted price for the 64K is now £69 (including VAT).

2) Is it misleading to call our large pack a 64K?
Well, it certainly contains 64K of RAM (and we don't know of any 56K chips!) The 64K of RAM is usable by any Z80 processor that will address it. Of course the ZX81 cannot do that but we do know of processors which do. The point I suppose is that MEMOTECH has been aiming for maximum forward compatibility. The 64K pack might for example come to be used (with paging and modifications) with the Spectrum. Another possibility is that it could be combined with a disc operating system for use with the ZX81. And so on.

3) Advantages of buying a MEMOPACK 16K.
Tim Langdell writes that all 16K packs use the triple rail 4116 chips but in fact the MEMOPACK 16K uses the single rail 5v HM 4816 chips. Incidentally, the latest version of our 16K pack will now work with either a MEMOPAK 32K, a Sinclair 16K pack or another MEMOPAK 16K to give a total of 32 or 48K.

4) What can you do where?
There are many misconceptions floating around (some have been shared by us) about how ZX81 addressing works. As a program is entered, it dynamically builds up the instruction file and array file (with the display file tucked in between). It is easy to get (with a small set of instructions) arrays of 45K or more with our 64K pack. On the face of it, however, it seems the display file has to stay below the 32K address mark and thus limits the instruction file to about 15K; but Michael Sims of Monikie, nr Dundee, has invented a neat dodge to give you up to 31K of instruction file! MEMOTECH in fact gave him a free HRG pack for discovering his method, which is summarised below:

HOW TO EXPAND YOUR INSTRUCTION FILE TO 31K OF INSTRUCTIONS

A) Instruction files *can* expand above the 32K mark provided the display file is never allowed to straddle addresses 32/67/32768 during input.

B) The solution is to check from time to time the value in the VARS variable, as this lies just above the end of the display file:
PRINT PEEK 16501 * 256 + PEEK 16400.

C) When the values of the VARS approaches 32767, enter a huge line into the program, like,
LET ZERO = 0 + 0 + 0 + 0 etc
with about 100 repetitions of (+ 0). This will push the display file entirely above 32768.

D) Check D-FILE to make sure that the display now begins above 32768:
PRINT PEEK 16397 * 256 + PEEK 16396.

E) The new maximum address for instructions will now be 49151. Going beyond this mark would entail hardware adaptation.

F) See chapters 26-28 of the Sinclair ZX81 Programming Manual for details of the way the system variables work.

5) Why put a Rolls/Royce add-on onto a Model T?
The fact is lots of people do! Our order list includes major oil companies, county councils and the logistics controllers of one of the largest air forces in the world. The ZX81 is not just for home and hobbies; it is not just for small businesses; it's used whenever someone in industry is in a tight spot and the systems department is too busy, too remote or too centralised.

David Jay, Memotech



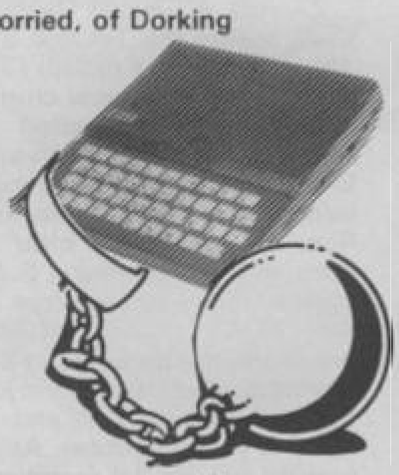
Funny people, these computerists

Dear ZX Computing,
I live on my own, so sometimes I only have my ZX81 for company. At first, I used to enjoy talking to the computer, but now I am worried since our relationship has become deeper and deeper.

Is there any danger of little ZX81's appearing? If there is, do I see a doctor or an electronics engineer?

I have been taking the usual precautions, such as switching off the power before going to bed at night.

Worried, of Dorking



Happy anniversary

Dear ZX Computing,
I just had to let you know that today is my 3 month anniversary — yes 3 whole months since I placed a telephone order with Sinclair Research for a 48K Spectrum. In the "early days" I was offered varying reasons for the delay, and now Sinclair Research inform me delivery could be a further "anything up to four weeks, sir"!!!

I offer no prizes for guessing how I feel about that situation.

I have always had a great deal of respect for Sinclair Research — an example of "the better side of British industry": that respect diminishes daily.

I remain a firm believer in the Spectrum and the thought of the Micro Drive keeps my order with Sinclair — for the moment.

However, I would like to offer one warning to Sinclair, if I may be so bold. Japanese industry has shown its abilities in both our motor cycle and motor car markets. It will not be very long before they enter our home computer market, and if the past performance in terms of delivery and back-up service to the consumer are not greatly improved upon by Sinclair Research, I believe they will very quickly become an example of the "best forgotten side of British industry".

P E Bloxham,
East Leake



No, no, no

Dear ZX Computing,
I am amazed that you should publish the 'hints'n'things' from James Higgs. I should *not* connect your Sinclair to a hifi system. The computer is designed to detect and decode the low voltage signal produced by the average portable tape recorder.

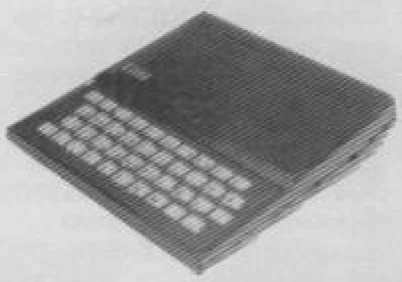
Feeding in the more powerful signals from a hifi deck will, more than likely, ruin the computer.

His second tip is also precarious. If switching off a lamp near the set produces a radio signal of sufficient power to affect the computer, it is most likely that severe arcing is taking place in the switch. The switch is dangerous and should be replaced.

Victor Wilson, Wigston

Thanks for your comments, especially on the latter point. I am not sure about your first claim, however, I have — despite advice to the contrary from Sinclair — always loaded my ZX80 from my music centre. It is the only recorder powerful enough to actually get it to load. The

computer is still working, two and a half years later, and it does not seem to have been harmed. I use a hefty semi-portable recorder, at three-quarters of maximum volume (about enough to fill a football stadium) to get a reliable load on my Spectrum. I guess the output of that is pretty close to the output from a hifi system. Again, the Spectrum appears to be working, and I have no problems with loading. T.H.



Business hints

Congratulations on the most recent issue. This has the making of the best ZX 81 Mag. to date, in that it gets away from the interminable games and into more practical uses.

As a small business first time user, with only a beginners course behind me, I find the time to delve deeper somewhat limited by the plethora of paperwork, which I hope the computer will eventually do something about. Thus any short cuts via your magazine will be greatly appreciated.

A couple of aspects which may be of use to some readers:

1. Having fitted a d'k keyboard; unlike most of the keyboard illustrations, I have filed a slot on the right hand side of the board and taken the strap out sideways, thus still allowing the Sinclair keyboard to be used either for games or dual instruction purposes.

2. In purchasing 1 metre Taping Leads (allowing the Tape Recorder to be placed on the floor when not actually in use), I obtained 2 3.5mm leads with different size mouldings; this ensures tht the leads do not get crossed.

Many thanks for ZX Computing. Hope you can keep up the quality,

C Flogdell, Brandon

Thanks for the comments, we're doing our best to keep up the standard.



Club roundup

Dear ZX Computing,
With interest in ZX81 and Spectrum computing running so high, it seems to me a shame that the Thames Valley area, (Slough, Reading, Windsor, Bracknell) does not have an active user group. I propose to start a group where enthusiasts can swap ideas, advice, programs etc, and would be pleased to hear from any of your readers who are interested in meeting other users on a regular basis. If anyone would like to contact me, my telephone number (evenings and weekends) is Maidenhead (0628) 21107.

Richard Shepherd

Dear ZX Computing,
I'd be pleased if you could include the following information in your magazine.

ZX-AID, a Sinclair Users Club, meets the 1st and 3rd Thursdays in the month (7.15 to 9.45pm) at a local church hall. Members are invited from the Walsall and West Bromwich areas. For further information contact Conrad Roe, Walsall 25465 after 6pm or write enclosing S.A.E. please, to 25 Cherry Tree Ave. Walsall, WS5 4LH. We are in the happy position of having a meeting room with both 13 amp sockets and heating for the Winter. At present we are 18 members but growing day by day.

My wife can put up with my correspondence and telephone calls now that meetings are away from our cramped Dining Room.

The club caters for beginners and the more experienced members alike, and we try programs, compare notes, share problems, give advice; we have plans for competitions, machine code courses and hardware projects; we have started to obtain concessions for shows and products, and maintain a Software and Hardware folder.

Conrad Roe.



When is 16K not?

Dear ZX Computing,
Congratulations on your magazine, which is a much better-balanced publication than all the 'yet another 1001 games which won't run' at present on the market. You have, so far, presented a balanced range of game programs, 'utility routines' and other, more theoretical articles. May I suggest, though, that program contributors give some estimate of the actual memory usage of their programs, for those (such as myself) who have available RAM greater than 1K but less than 16K.

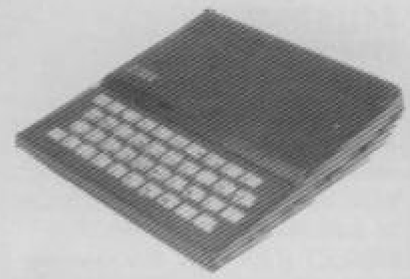
Many of your so-called "16K programs" will in fact run in less than 16K (Dragons Gold in the Aug/Sept issue, for instance, *just* fits into 3K). PRINT PEEK 16396 + 256 PEEK 16397 − 16509 gives the length of the program, without, of course, taking into account memory used by variables and the display file.

I should also be grateful if you can pass on the following useful piece of information:

As is well-known, if available RAM is less than 3¼K, the display file is not full-sized, causing some spectacular crashes if values are POKEd into the display file. Perhaps not so well-known though, is the fact that, after initialisation, the only function of the system variable RAMTOP is to determine the status of the display file. If RAMTOP is set to greater than 3¼K, therefore, the computer always produces a 'padded-out" display file. The statement POKE 16389, 77 will ensure this. POKE 16389, 76 (or less, minimum 67) undoes this.

Obviously, if only 1K of RAM is available, the full display file leaves little space for programs, but the procedure above may be of use to users with 1 or 2K expansion RAM fitted.

Anthony Walton, Burn, Selby



'Sabature' replies

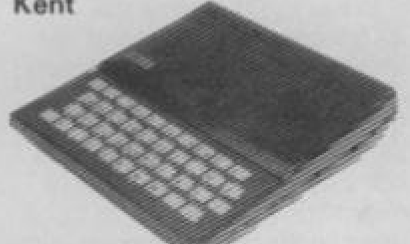
Dear ZX Computing,
Firstly, well done on a good publication.

Secondly, concerning your reviews of STARTRECK.

The bug in the photon torpedo routine of the Macronics game (they market it for me) should have been removed from sale tapes some time ago. For readers who have the bug, it is cured by adding the line 5085 LET K9 = K9 − 1.

Phil Garratt is correct that a 'Sabature' got to the program.

V Vaughan,
J W V Software, Strood, Kent



Found — the answer

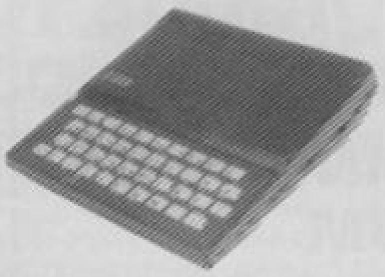
Dear ZX Computing,
You may be interested in two cures which I have found for two major problems that I have been experiencing, i.e. firstly to stop my 16K RAM pack "crashing" — all I have done is unscrewed the 4 screws of its casing, which allows the casing to come away in two halves, and with the casing removed and the two inside circuit boards bent slightly further apart than the position in which they are held when fitted inside the casing — this has completely eliminated any further crashings, regardless of how long the computer is used for.

The other problem is that of LOADING — the cure was simply to move my television

set as far away as possible from both the ZX81 and my cassette player and then to completely turn the T.V. set around whilst SAVING or LOADING; thus, with the set's back towards, and at a distance from my cassette player, there are no further problems.

A P Gower
Thorpe Bay, Essex

WANTED!



More anniversaries

Dear ZX Computing,
Having a high regard for Clive Sinclair's design skills, I would like to enthuse over his Spectrum — but eleven weeks after ordering one I still haven't even got a delivery date! The advertisement in the current issue of ZX Computing is therefore particularly ironic. I am sure that this is a larger problem than Sinclair would care to admit. When will his company match design skill and marketing claims with production? Would it make more sense to postpone marketing announcements until orders can be met? It's almost enough to drive me to a VIC-20!

Ian Black, Darlington

And more

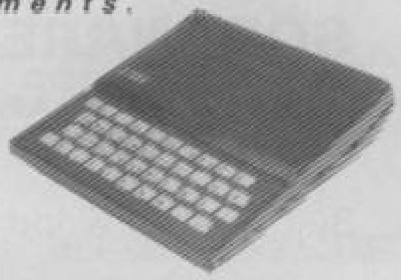
Dear ZX Computing,
After purchasing ZX Computing last month I have to agree with you that it is not only the biggest magazine for the Sinclair user but also by far the best.

However I would like to criticise Sinclair, and anybody awaiting delivery of a Spectrum will no doubt be inclined to agree. Whilst Clive Sinclair is hailed as the man who brought personal computing to the man on the street, I doubt whether he will be remembered as a production engineer.

I have been waiting, (despite several phone calls), for my Spectrum for over 7 weeks as against the claim for delivery of 28 days. I have been assured that my cheque has not been cashed but put into a so-called customer account where upon it will be cashed on dispatch of my order. However, this account is no doubt earning interest and if one considers the size of its balance it appears to me that 'Uncle' Clive has, what amounts to a free deposit acount. Come on Clive, sort it out.

A Wiater, London, N10

I rang Sinclair Research in the second week of October regarding Spectrum deliveries, and was told that — at that stage — Spectrum deliveries were 10 weeks (16K) and 12 weeks plus (48K), but now that the company had decided to have units constructed by a second firm (EMI), the delivery period was expected to come down rapidly. He said those actually ordering in the second week of October could expect delivery within six weeks, as the initial flood of orders had been all sent out, and by early November deliveries should be close to the 28 day period promised in the advertisements.



The 10,000th prime

Dear ZX Computing,
May I begin by saying that you are the best magazine on the market for the Sinclair owner?

Your story "Getting primed" prompted me to find the ten-thousandth prime. I had a prime generator program (better, in my biassed opinion, than yours!) which I modified to count as it went; it is called "PRIMES AD NAUSEAM".

It runs in FAST and you use BREAK and CONT to see what is going on. You could use it in SLOW but not, perhaps, if you expect to live for the usual three-score years and ten . . .

Be that as it may, I cheated by using the figures in your story and substituting them in lines 30 and 32. Like you I was concerned about the heat which builds up ("Buy a Sinclair and fry eggs while you compute") so every so often I stopped it, and substituted the highest numbers I'd so far reached when re-starting.

It took about three hours to find Prime 10,000 which is 104,683.

Another program I've written, "PRIME FINDER", tells you whether or not a number is prime. I ran 104,683 into it, not without some trepidation, and it confirmed the prime-ess (if that's the word) in about six seconds.

Once again, well done, and may there be many more issues to come.

Andrew Turek,
London, NW11

May I point out that it is not compulsory to start letters saying that ZX Computing is the best computing magazine for Sinclair owners in the country? However, although it is not compulsory, it helps sooth staff members with feelings of insecurity and inferiority (such as Tim H) and certainly increases the chance that we'll print your letter!.



More pigs

Dear ZX Computing,
After studying the pig-Latin program which appeared in the second issue of your magazine, I wrote the enclosed program which converts pig-Latin to English. It runs comfortably on a 1K ZX81. By the way, great mag. Keep it up.

```
40  PRINT "ENTER MESSAGE WORD FOR WORD"
50  PRINT "ENTER $ TO STOP"
60  PRINT
70  INPUT A$
80  IF A$ = "$" THEN STOP
90  LET B$ = A$ (TO(LEN A$) - 2)
100 LET A$ = A$ (LEN A$ - 1) + B$
110 PRINT A$;" ";
120 GOTO 70
```

Alexander Rogers, Radlett

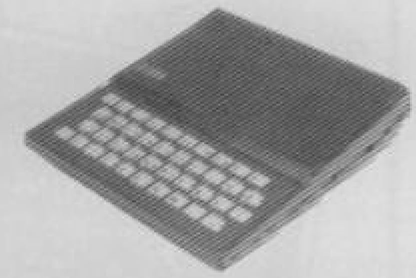


Wobble, wobble

Dear ZX Computing,
I read in your magazine yet another letter on the problem of Sinclair RAM packs and program crashes.

The reason I am writing is to suggest that nothing needs to be done to the ZX81 and RAM pack, except to raise the '81 off the operating surface by 1/8-3/16" or so (on a piece of hardboard, for example), leaving the pack 'floating free', held only by the edge connector.

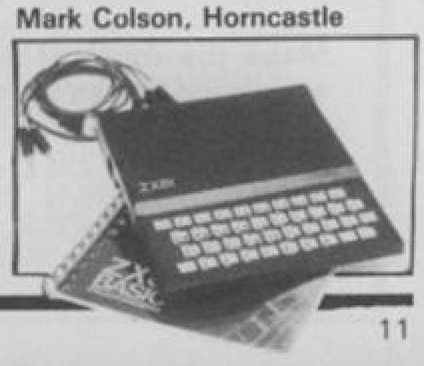
The above method cures the problem by removing the differential movement between the ZX81 and the RAM pack, which occurs when both are on the same flat surface, and the ZX81 keyboard is operated. The pressure pushes the computer down on its rubber feet, but the RAM pack has solid footings, and cannot accommodate the movement. The result is a momentary break on one or more contacts of the edge-connector . . . and operator frustration.
V J Ludlow, Weston-super-Mare



Manual mistakes

Dear ZX Computing,
Have you noticed that CHR$ 7 and 135 are printed as the same character in the Sinclair manual? Also, the graphic character on key 3 is missing? The missing character has the code 135,

Mark Colson, Horncastle



A PAIR OF ZIPPY ROAD RUNNERS

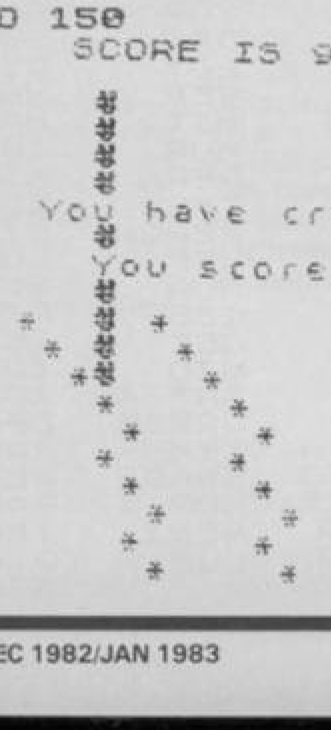


ZX81 programs often run, just as they are, when typed into a Spectrum. But usually the effort of adding a little colour and sound is worthwhile.

For this article, Tim Hartnell wrote a ZX81 'car driving' program in which you use the "Z" and "M" keys to drive it down a twisting, turning track; and then decided to convert the game for the Spectrum.

You might like to run Tim's versions first, and then work out your own variations.

```
  10 REM ROAD RUNNER
  20 LET T=0
  30 LET A=10
  40 LET X=13
  50 LET Y=12
  55 LET K=INT (RND*2)
  60 LET A=A-(K=1 AND A>1)+(K=0
AND A<24)
  70 PRINT AT Y,X-1;"V"
  80 SCROLL
  90 PRINT TAB A;"▒";TAB A+5;"▒"
  95 PRINT AT Y+1,X-1;
 100 IF PEEK (PEEK 16398+PEEK 16
399*256)=128 THEN GOTO 140
 110 LET X=X-(INKEY$="Z" AND X>2
)+(INKEY$="M" AND X<32)
 120 LET T=T+1
 130 GOTO 55
 140 PRINT AT Y,X-1;"V"
 150 PRINT AT 6,8;"YOU HAVE CRAS
HED"
 160 PRINT AT 8,10,"YOU SCORED "
;T
 170 PRINT AT 6,8;"YOU HAVE CRAS
HED"
 180 GOTO 150
```




```
  10 REM ROAD RUNNER
  20 LET C=0
  30 LET T=0
  40 GO SUB 250
  50 LET A=10
  60 LET X=13
  70 LET Y=12
  80 LET K=INT (RND*2)
  90 LET A=A-(K=1 AND A>1)+(K=0
AND A<24)
 100 REM NEXT LINE CONTAINS A
         GRAPHIC C AS DOES 200
 110 PRINT AT Y,X-1; INK 1;"▒"
 120 PRINT AT 20,A; INK 2;"*";TA
B A+5;"*"
 130 PRINT
 140 POKE 23692,-1: PRINT
 150 PRINT INK 6; PAPER 2;AT 0,1
0;" SCORE IS ";T;"  "
 160 IF SCREEN$ (Y+1,X-1)="*" TH
EN GO TO 200
 170 LET X=X-(INKEY$="Z")+(INKEY
$="M")
 180 LET T=T+1
 190 GO TO 80
 200 PRINT AT Y,X-1; INK 1;"▒"
 210 PRINT AT 6,8; FLASH 1; BRIG
HT 1;"  You have crashed!!  "
 220 PRINT AT 8,10; FLASH 1; BRI
GHT 1; INK RND*7; PAPER 9;"  You
 scored ";T;"  "
 230 BEEP .01,RND*20-RND*20
 240 GO TO 210
 250 FOR J=0 TO 7
 260 READ Z
 270 POKE USR "C"+J,Z
 280 NEXT J
 290 RETURN
 300 DATA BIN 00110110,BIN 00110
110,BIN 00111110,BIN 00010100,B
IN 00111110,BIN 00110110,BIN 00
011100,0
```

TALKING TURKEY

This program has decided that although the English language does not exist, some of its rules do.

Using these rules, the computer tries to invent English words, and manages to do so surprisingly often. Around seven per cent of the output of this program — written by Paul Holmes — should be real words. The ZX81 uses knowledge of the frequency of occurence of certain letters in words in English to dictate how often the letters are used in creating randomly generated words.

If you leave this program running for a million years it may well write the Gettysbury Address ('Three score and seven years ago our fathers founded . . .').

PROGRAM LISTING

```
 10 DIM A$(26,23)
 20 LET A$(1)="TNIRSHMGBCDFJKLP
QUAVWXYZ"
 25 LET A$(2)="EAOIU"
 30 LET A$(3)="ETAOISHU"
 35 LET A$(4)=A$(3)
 40 LET A$(5)="ETANRSHMGBCDFJKL
PQUVWXYZ"
 50 LET A$(6)=A$(2)
 60 LET A$(7)="EAOISHU"
 70 LET A$(8)=A$(2)
 80 LET A$(9)="TEONRSHMGBCDFJKL
PQUVWXYZ"
 90 LET A$(10)=A$(2)
100 LET A$(11)=A$(2)
110 LET A$(12)=A$(2)
120 LET A$(13)=A$(2)
130 LET A$(14)=A$(2)
140 LET A$(15)="TAONIRSHMGBCDFK
JLPQUVWXYZ"
150 LET A$(16)=A$(2)
160 LET A$(17)="U"
170 LET A$(18)=A$(2)
180 LET A$(19)="EAOIHU"
190 LET A$(20)=A$(2)
200 LET A$(21)="AO"
210 LET A$(22)=A$(2)
220 LET A$(23)=A$(2)
240 LET A$(24)=A$(2)
250 LET A$(25)=A$(2)
260 LET A$(26)=A$(2)
265 LET L=INT (3+RND*INT (RND*4
+1))
270 LET X=INT (RND*26+1)
280 FOR I=1 TO L
290 PRINT CHR$ (X+37);
300 LET C=CODE A$(X,(INT (RND*I
NT (RND*23)+1)))
310 IF C=0 THEN GOTO 300
320 LET X=C-37
330 NEXT I
340 IF INKEY$="" THEN GOTO 340
345 CLS
350 GOTO 265
```

A 'poetry' version of the program:

```
  5 REM "POETRY" VERSION
 10 DIM A$(26,23)
 20 LET A$(1)="TNIRSHMGBCDFJKLP
QUAVWXYZ"
 25 LET A$(2)="EAOIU"
 30 LET A$(3)="ETAOISHU"
 35 LET A$(4)=A$(3)
 40 LET A$(5)="ETANRSHMGBCDFJKL
PQUVWXYZ"
 50 LET A$(6)=A$(2)
 60 LET A$(7)="EAOISHU"
 70 LET A$(8)=A$(2)
 80 LET A$(9)="TEONRSHMGBCDFJKL
PQUVWXYZ"
 90 LET A$(10)=A$(2)
100 LET A$(11)=A$(2)
110 LET A$(12)=A$(2)
120 LET A$(13)=A$(2)
130 LET A$(14)=A$(2)
140 LET A$(15)="TAONIRSHMGBCDFK
JLPQUVWXYZ"
150 LET A$(16)=A$(2)
160 LET A$(17)="U"
170 LET A$(18)=A$(2)
180 LET A$(19)="EAOIHU"
190 LET A$(20)=A$(2)
200 LET A$(21)="AO"
210 LET A$(22)=A$(2)
220 LET A$(23)=A$(2)
240 LET A$(24)=A$(2)
250 LET A$(25)=A$(2)
260 LET A$(26)=A$(2)
262 FOR G=1 TO 100
265 LET L=INT (3+RND*INT (RND*4
+1))
270 LET X=INT (RND*26+1)
275 SCROLL
277 FOR H=1 TO RND*5+2
280 FOR I=1 TO L
290 PRINT CHR$ (X+37);
300 LET C=CODE A$(X,(INT (RND*I
NT (RND*23)+1)))
310 IF C=0 THEN GOTO 300
320 LET X=C-37
330 NEXT I
332 IF L<5 THEN LET L=L+INT (RN
D*2)-INT (RND*2)
335 PRINT " ";
336 NEXT H
337 IF RND>.8 THEN SCROLL
340 NEXT G
350 GOTO 262
```

Part of the output:

```
AIBO      YOP     AMA
DTEH      DEN     SUABA
TOO       LOT     DTAKA
UOTI      VEE     HACA
NEFAIT    NIT     MAT
REE       HUA     INAX
HUAH      RERA    XIO
BIE       TEGO    SOM
YINOR     PEAGH   SITE
YIE       WEF     LUOO
JEC       NUA     SEX
BATU      IEB     PIG
JIT       CET     ZEXE
RIH       NIO     ZOT
OGI       SAT     ZOGAN
RAC       UARIB   BED
IES       TEM     XAN
KITE      SHAG    FENE
LIO       KOR     BAUAN
BAHETA    HEMI    TAMA
KASITA    LUO     DTAC
LU                HAC
```

Stretching your Spectrum

Dilwyn Jones, who has written several articles on making the most of the ZX81, now turns his attention to the Spectrum. This article will show you how to get the best out of your Spectrum.

Screen tricks

Enter and run this program. What does it do?

```
  10 DIM i$(704)
  20 PRINT AT RND*20,RND*31,CHR$
 (RND*223+32)
  30 PRINT AT 0,0; OVER 1; INVER
SE 1;i$
  40 GO TO 20
```

About twice a second something is printed on the screen, then the entire screen is inverted. Who needs machine code? Actually it's done by printing a screenful of spaces OVER the entire screen in INVERSE which has the effect of causing everything that was white on the screen to turn black and everything that was black to turn white: normally you would expect OVER to use its EXOR action to erase some parts, but there isn't anything to EXOR within a string of spaces, so it can provide a true screen inversion very quickly. This works well in black and white, but it is easy to modify for colour by adding local PAPER, INK, FLASH and BRIGHT controls with an 8 parameter each (to prevent global colour parameters etc. to play havoc under some circumstances). All this does is ensure that the same attributes are maintained, but that INVERSE 1 is effected.

```
  10 DIM i$(704)
  15 FOR i=1 TO 50
  20 PRINT AT RND*20,RND*31; INK
 RND*7; PAPER RND*7; FLASH RND;C
HR$ (RND*223+32)
  25 NEXT i
  30 PRINT INVERSE 1;AT 0,0; OVE
R 1; PAPER 8; INK 8; BRIGHT 8; F
LASH 8;i$
```

The same idea can be used to turn all text and graphics on the screen a particular colour by omitting the INVERSE 1 statement (or specifying INVERSE 0) and specifying an INK colour rather than leaving it INK 8. For instance, this program writes random characters on the screen in random INK and PAPER colours, for demonstration, then changes all characters to black and while keeping brightness, flashing and paper attributes the same:

```
 10 DIM i$(704)
 15 FOR i=1 TO 50
 20 PRINT AT RND*20,RND*31; INK
RND*7; PAPER RND*7; BRIGHT RND;
FLASH RND;CHR$ (RND*223+32)
 25 NEXT i
 30 PRINT AT 0,0; OVER 1; PAPER
8; INK 8; BRIGHT 8; FLASH 0;i$
```

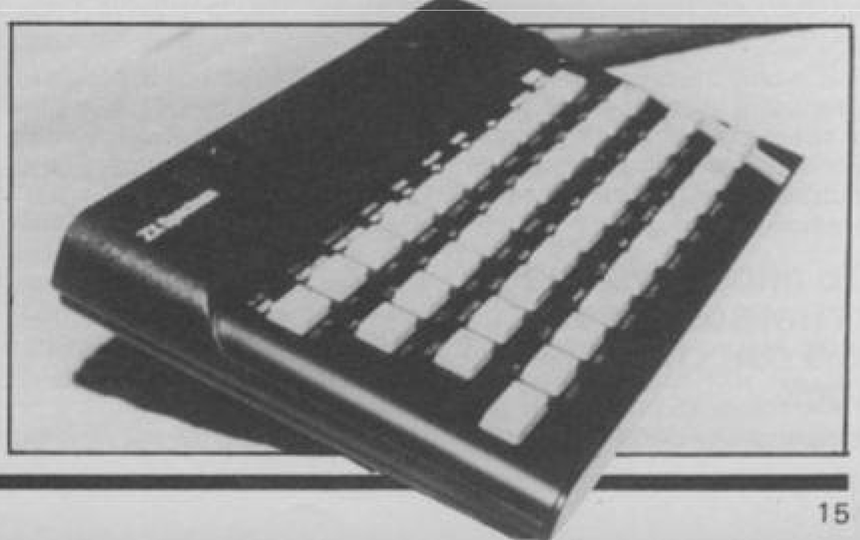
You may have noticed that some INKs and PAPERs come out the same after the random printing in line 20. This is a common problem. Problem? No! Just specify INK 9. You can now read everything.

```
 10 DIM i$(704)
 15 FOR i=1 TO 50
 20 PRINT AT RND*20,RND*31; INK
RND*7; PAPER RND*7; BRIGHT RND;
FLASH RND;CHR$ (RND*223+32)
 25 NEXT i
 30 PRINT AT 0,0; OVER 1; PAPER
8; INK 9; BRIGHT 8; FLASH 8;i$
```

We can do the same to the PAPER. By specifying the paper colour, and leaving all other attributes the same, the entire background colour can be changed without disturbing anything on the screen or using CLS. Note that anything written in this colour on the screen may appear to vanish as, say, green text on green paper is not all that easy to read! This example draws random characters with random attributes, then sets the entire background to yellow.

```
 10 DIM i$(704)
 15 FOR i=1 TO 50
 20 PRINT AT RND*20,RND*31; INK
RND*7; PAPER RND*7; BRIGHT RND;
FLASH RND;CHR$ (RND*223+32)
 25 NEXT i
 30 PRINT AT 0,0; OVER 1; PAPER
6; INK 8; BRIGHT 8; FLASH 8;i$
```

You get an interesting effect with any yellow area that has a BRIGHT attribute of 1 with the above program. If you had provided user prompts in BRIGHT 1 or FLASH 1 (ie. extra bright or flashing) to highlight them, then after they had been acted upon you wished to cancel them; et voila . . .



to turn off bright spots:

```
 10 DIM i$(704)
 15 FOR i=1 TO 50
 20 PRINT AT RND*20,RND*31; INK
RND*7; PAPER RND*7; BRIGHT RND;
FLASH RND;CHR$ (RND*223+32)
 25 NEXT i
 30 PRINT AT 0,0; OVER 1; PAPER
8; INK 0; BRIGHT 8; FLASH 8;i$
```

to turn off flashing:

```
 10 DIM i$(704)
 15 FOR i=1 TO 50
 20 PRINT AT RND*20,RND*31; INK
RND*7; PAPER RND*7; BRIGHT RND;
FLASH RND;CHR$ (RND*223+32)
 25 NEXT i
 30 PRINT AT 0,0; OVER 1; PAPER
8; INK 8; BRIGHT 0; FLASH 8;i$
```

Note that in all the above examples, the "screen tricks" are all accomplished in one line! Remember: The answer to the ultimate question of life, the universe and everything is a string of 704 spaces printed OVER 1 over the entire screen with colour 8's!

This technique opens up an interesting possibility — if you want to draw a complex shape which would normally be very slow, first draw it the normal way in the same INK colour as the PAPER colour so that its invisible, then use the above technique to change the shape's colour so that it instantaneously (well, a fraction of a second) becomes visible. Try this program which draws 4 concentric circles in magenta on a yellow background. The drawing process takes about four seconds.

```
  5 INK 3: PAPER 6: CLS
 10 DIM i$(704)
 15 FOR i=10 TO 70 STEP 20
 20 CIRCLE 120,90,i
 25 NEXT i
```

This program initially draws the circles in yellow on a yellow background, then after drawing changes the colour of the circles to magenta on yellow. You have to stare at a blank yellow screen for a couple of seconds, but when they appear, the circles seem to be drawn almost immediately. In practice, you'd be able to disguise the delay in a program so that drawing appeared instantaneous, or the shape would be built up over a long time then quickly displayed when required.

```
 10 DIM i$(704)
 15 FOR i=10 TO 70 STEP 20
 20 CIRCLE 120,90,i
 25 NEXT i
 30 PRINT AT 0,0; INK 3; OVER 1
;i$
```

This is only the bare bones of an idea: using an overprinted string of spaces to control the display attributes without affecting the picture is a powerful, fast programming tool.

To allow the spectrum to switch off/on the caps lock

Since the FLAGS2 System Variable 23658 has BIT 3 set to indicate CAPS LOCK ON, a simple POKE 23658,8 will switch on CAPS LOCK and POKE 23658,0 will turn it off again. Care is necessary when POKEing into this system variable since it controls other facilities as well, e.g. BIT 6 controls OVER. For fun, plug in your printr and enter this;

POKE 23658,2

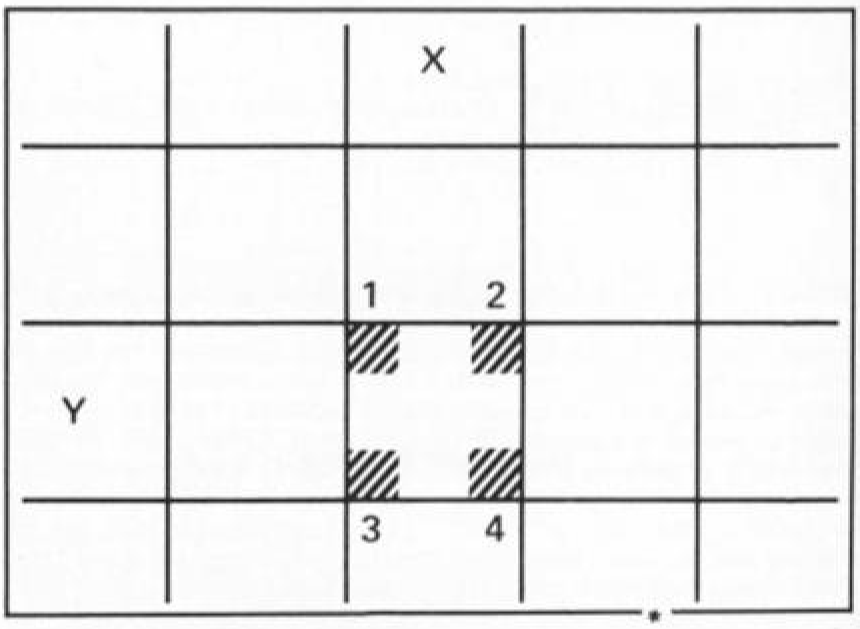
Don't worry: you won't break it or waste paper, although you will wake it up. However, back to the point. The facility to POKE the caps lock on and off may be useful in programs which require the operator to press certain keys — normally some law somewhere would say that if you'd arranged for a Y or N answer for yes or no, the CAPS LOCK would be off and you'd press y or n, just to confuse matters. This is how it could be used to detect capitals for a Y or N (Yes or No) response:

```
1000 PRINT "Do you want
another game (Y or N)?"
1010 POKE 23658,8
1020 IF INKEY$ = "Y"
THEN RUN
1030 IF INKEY$ = "N"
THEN STOP
1040 GOTO 1020
```

Address 23658 would normally have a value of 0 but it may be wise to check if you're using OVER etc. in a program and ensure that you only affect BIT 3.

Matching up print and plot co-ordinates

Suppose you wanted to PRINT AT Y,X;



The PLOT co-ordinates corresponding to the four corners 1 to 4 of character cell Y,X (shaded in above diagram) would be in PLOT X,Y format:-

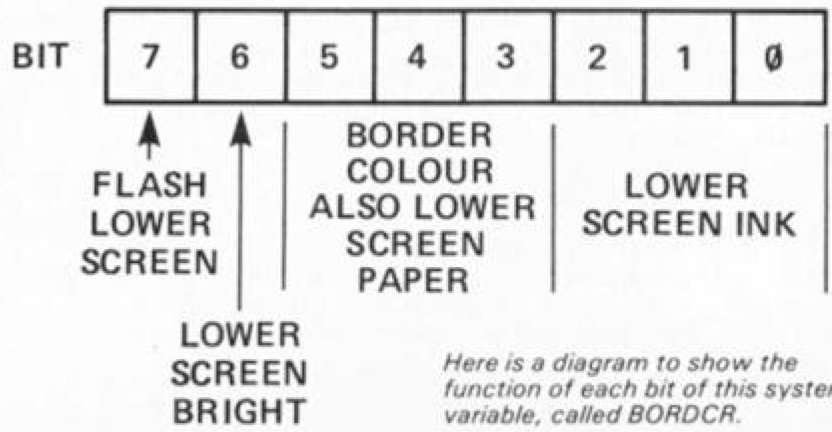
(1) X*8, (21-Y)*8 + 7
(2) X*8 + 7, (21-Y)*8 + 7
(3) X*8, (21-Y)x8
(4) X*8 + 7 (21-Y)*8

From these you should be able to work out the positions of all pixels within the print position should you require to draw a line through a known print position.

Lower screen attributes

Normally you cannot change the attributes of the lower screen, except for the PAPER colour, which follows the border colour.

Would you like a green report code and cursor? Neither would I, but it can be done. System variable 23624 contains the attributes used for the lower screen and the border colour.



Here is a diagram to show the function of each bit of this system variable, called BORDCR.

By pokeing various values into this system variable you could achieve, say, a flashing black and white lower screen or a white lower screen that stands out brighter than the rest of the white screen for INPUTs etc. Try these two with a white screen (PAPER 7:CLS)

POKE 23624,BIN 10111000 (184)
POKE 23624,BIN 01111000 (120)

You cannot normally get this effect with INPUT statements, as colour controls etc. only affect the prompt string. Changing BORDER colour affects this: for instance lower screen INK (being "automatically" 9) would revert to either white or black to ensure maximum contrast so that anything typed in the lower screen can be read easily.

Pause and FOR/NEXT loops

There is normally no problem with using PAUSE on the Spectrum. But where a fixed delay is required, PAUSE may cause problems. PAUSE is cut short by a keypress so if you held your finger on a key all PAUSES would never occur. This problem can be alleviated by the use of FOR/NEXT loops as delay loops. To achieve a delay of 1 second (approximatly) use

FOR A = 1 TO 220:NEXT A

Non-deletable program lines

Wouldn't it be nice to insert a line like

10 REM (C) FRED BLOGGS 1982

into your program, knowing it couldn't be edited out and prevent other people copying that program without your author credit? Deleting the above line is easy: just type 10 followed by ENTER and the line has been deleted in the normal way. What is needed is a method of inserting lines into a listing which are very difficult if not impossible to delete. Part one of the answer is that if you manage to get a line numbered 0 into a listing it cannot be deleted in any of the normal ways since line number 0 is normally associated with direct commands (eg. enter the command PRINT without a line number: you should get report 0,0). If you attempted to enter

0 REM (C) Fred Bloggs 1982

you would be rewarded with the cheery message NONSENSE IN BASIC with report C. So that's out. What we have to do is enter a line with a normal line number (eg 10) then change this number to zero. Difficult? Not a bit (no pun intended). We could do this by looking through the program for the line number, followed by a REM then POKE away until we get what we want.

However, this would be too slow and messy. A better way is to use the system variable NXTLIN contained in 23637/8 which contains the address of the start of the next program line (NOTE: LINE not STATEMENT). The Spectrum manual tells us that each BASIC program line starts off with a line number stored in two bytes in the order More significant Byte (MSB) followed by less Significant Byte (LBS). Therefore, line 1 would be 0,1 and line 258 would be 1,2 (1x256 + 2). So if we POKEd 0 into both bytes we'd get our objective of a virtually undeletable program line. Here's how to do this in two lines of BASIC:

1 LET a = PEEK 23637 + 256 ×PEEK 23638: POKE a,0: POKE a + 1,0:STOP
2 REM (C) Fred Bloggs 1982

```
   1 LET a=PEEK 23637+256*PEEK 2
3638: POKE a,0: POKE a+1,0: STOP

   2 REM © Fred Bloggs 1982
```

RUN the program. Now LIST the program. Note the zero line number where there used to be a 2, and note also how the lines are not sorted into the correct order; sorting only takes place when lines are entered; once they're in they stay in order although subsequent lines will go in the right place. Line 1 is no longer needed — delete it as normal to prevent others using it to correct what you've done. You should now have:

0 REM (C) Fred Bloggs 1982

```
0>REM © Fred Bloggs 1982
```

Try deleting it by typing in its line number; try using EDIT. Quite secure isn't it? To delete it you will have to go through all that POKEing again. But if you think about it you've got a problem — you can't use the system variable NXTLIN because line 0 is now the first program line — any other lines entered are sorted as entered into numerical order and will all go after line 0. NXTLIN will not give the right address unless used in the line before — tough luck. For security (!) I'll leave you to work out how to delete line 0. There are several ways of doing this, but they might not be very obvious, or very "roundabout" ways. There are no prizes for doing this as it is not intended to be done. You could place all this into any part of a program and, if you're keen enough, you could place a bright, flashing, coloured copyright statement onto each page of a listing so it stands out whichever part of a listing is viewed. If you aim to use it a lot, you could put the two lines on tape and use MERGE to add it to your program. If starting programs from scratch, save the two lines on tape using

SAVE "(NAME)" LINE

This will automatically create line 0 when reloaded, leaving only the task of deleting line 1.

Press any key to continue

A common requirement is to suspend execution of a program pending an instruction from the operator. An example would be displaying a list of instructions then ask the operator to press any key to continue after finishing reading the instructions. This part of the program may well look like this:

```
.....(instructions)

1000 PRINT "Press any key to con
tinue."
1010 IF INKEY$="" THEN GO TO 101
0

(rest of program)...
```

This is fine, but if you press CAPS SHIFT or SYMBOL SHIFT the program will ignore you and other people will remark "what a stupid program". There are some ways out of this:

```
1000 PRINT "Press any key except
 the shift keys to continue"
1010 IF INKEY$="" THEN GO TO 101
0

1000 PRINT "Press ENTER to conti
nue"
1010 INPUT A$
```

Incidentally, you may have noticed with the programs using INKEY$ that INKEY$ does not respond to either shift alone, but if both SHIFT keys are pressed simultaneously, the program continues. Pressing both shift keys simultaneously (as when you enter E mode) produces CHR$ 14. The above examples are fine, but wouldn't it be nice if we could truly press any key to continue? Any key of course meaning any of the forty keys on the Spectrum keyboard including both SHIFTs. Here is one way in which this could be done.

```
1000 PRINT "Press any key to con
tinue"
1010 IF INKEY$="" AND IN 65278=2
55 AND IN 32766=255 THEN GO TO 1
010
```

The keyboard is located in what is called I/O space, meaning INPUT/OUTPUT. These are methods of getting information in and out of the computer from and to the outside world. The MIC and EAR sockets, the internal loudspeaker, the keyboard, the printer and microdrives and the RS232 interfaces are all examples of I/O in action. The most significant

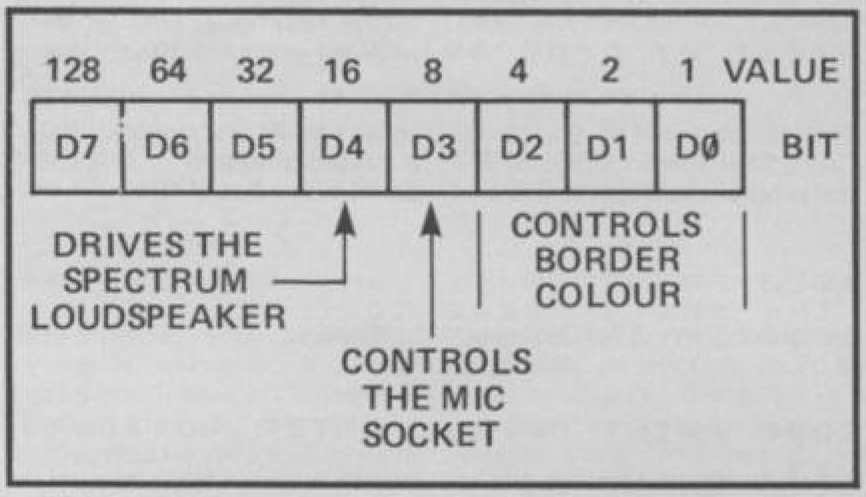
difference between memory addressing as far as the user is concerned is that PEEK and POKE only work with memory, be it RAM or ROM. The I/O commands IN and OUT are concerned with getting information to or from the computer from/to the outside world. There are 65536 of these IO ports, just as there can be 65536 memory locations, but they may or may not all be in use, just as all memory space is not used in a 16K Spectrum.

There are two commands in BASIC to handle the I/O ports. These are IN and OUT which can be thought of as working like PEEK and POKE respectively. Functions like INKEY$ also access the I/O ports, but making use of machine code equivalents of IN and OUT in their own little ways. The next question is how do you know which of these PORTS are used for what? Chapter 23 of the Spectrum manual outlines them briefly, but the ones most likely to be of use are those associated with the keyboard, at least at this stage.

As an example using OUT, let us play with PORT 254, which amongst other things sets the border colour and drives the loudspeaker. This can be demonstrated by running this short program:-

```
10 OUT 254,INT (RND*255)
20 GO TO 10
```

You should hear a clicking noise from the Spectrum's loudspeaker and see the screen's border colour go haywire! The colour changes so rapidly, you may be able to see several border colours at once! Note that whilst this program is running, the lower two lines of the screen do not change colour (they would normally be the same colour as the border). The border reverts to the colour of the lower screen when you type something. If you understand anything about Binary, this diagram of the eight bits of PORT 254 may help to explain how the port manages to do more than one thing at a time. Like a memory location I/O ports are eight-bit bytes.



DO, D1, D2 etc. mean bit 0, bit 1, bit 2 etc. The D usually stands for DATA, but that need not bother us now. Since, only bits 0 to 4 are used, we should have replaced line 10 in the previous program with

```
10 OUT 254,INT (RND*32)
```

since the bits used could add up to 0 (lowest) and 31 (highest).

More useful to us are the I/O ports that are associated with the keyboard. There are eight ports, each handling a row of five keys on the left or right half of the keyboard. For example PORT 61438 is associated with the row of five keys, 6 (bit 4) 7 (bit 3) 8 (bit 2) 9 (bit 1) and 0 (bit 0). Try this program which prints out the value of port 61438 over and over again. Try pressing keys 6 to 0 to see what effect it has. Press more than one key at a time.

```
10 PRINT IN 61438
20 PAUSE 100
30 GO TO 10
```

RUN it and see how it prints 255 all the time, unless you press one of the 5 keys in the half-row 6 to 0 on the keyboard. The numbers obtained may look pretty random, until you realise how the numbers are worked out. You may have realised that 255 is the value for 'no key pressed''. You may also know that 255 in binary is 11111111. So since numbers obtained when keys are pressed are less than 255, can you imagine that pressing a key turns one of those binary ones into a zero?

Study this diagram, which shows which bits of which ports relate to which keys. In particular, try to study those keys we've been using as examples, 6 to 0.

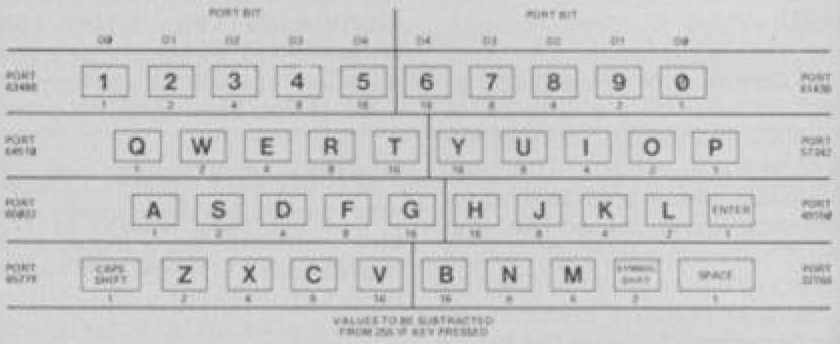


Diagram of I/O ports associated with the keyboard, and which bit of the ports are associated with each key. Note how bit 0 is always on the outside and bit 4 on the inside.

RUN the program again and every time you press a key, subtract the number written under the keyboard keys in the diagram from 255, eg. if you're pressing 0 subtract 1 from 255, giving 254. If you're pressing 8, subtract 4 from 255, giving 251 and so on. You should get the same number as that the program writes on the screen.

This may not make much sense at the moment, but persevere and hopefully all will become clear in due course. Written above the keys in the diagram are the symbols D0 to D4 again — these represent individual bits of the I/O port. In this application only bits 0 to 4 are used for the keyboard, as there are only five keys to be checked per port. Let us have some simple examples to demonstrate a simple use of IN to scan the keyboard:

to check if the R key is pressed

```
IF IN 64510=(255-8) THEN PRINT
"R is pressed"
```

to check if the Y key is pressed

```
IF IN 57342=(255-16) THEN PRINT
"Y is pressed"
```

to check if the SPACE key is pressed

```
IF IN 53742=(255-1) THEN PRINT
"SPACE is pressed"
```

Of course you need not write the expression in brackets in full like the examples above — they've only been written in full to illustrate the point that you subtract the bit value from 255. Note that if you add up the bits' values all together, the answer is the same. At this stage it does not make much difference how you do it. Getting a correct result and understanding it is most important now. The essential thing is to note that any bit is only a zero if the corresponding key is pressed. This explains how you get a value of 255 if nothing is pressed — all bits are 1, so the total is 255 in decimal. Take the example of the K key being pressed. The I/O port associated with that half-row of 5 keys is 49150 (see keyboard diagram). Each byte or PORT has eight bits, like this:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3
1	1	1	1	1

BIT 2	BIT 1	BIT 0
1	1	1

The above shows the half-row with no keys pressed. When the K key is pressed, this is how the port looks.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3
1	1	1	1	1

BIT 2	BIT 1	BIT 0
0	1	1

It now has a value of BIN 11111011 which is, (in decimal) (255 − 4) or 251, which is also the same as (128 + 64 + 32 + 16 + 8 + 2 + 1). Technically, adding up the bits individually is the correct way of doing it, but the other method also works for reasons we won't go into here, and it's generally easier to use for this application. You could do the same for any key on the keyboard. On its own,

```
IF INKEY$="K" THEN PRINT "K pres
sed"
```

achieves nothing over

```
IF IN 49150=251 THEN PRINT "K pr
essed"
```

However, there are advantages. You can check if either SHIFT key is pressed for example, which you couldn't do with INKEY$, eg.

```
IF IN 65278=254 OR IN 32766=253
THEN PRINT "SHIFT pressed"
```

INKEY$ also differentiates between upper and lower case letters so that IF INKEY$ = "k" THEN . . . is not the same as IF INKEY$ = "K" THEN. . . , whereas IF IN 49150 = 251 THEN . . . just checks if the k key is pressed, irrespective of whether CAPS LOCK or CAPS SHIFT is on. Using IN to scan the keyboard also allows us to check if more than one, or combinations of, keys are being pressed, eg.

```
IF IN 49150=(255-2-4) THEN PRINT
"K and L pressed"
```

One application for this would be in games where the cursor control keys are used to control movement on the screen in the direction of the arrows. Most games only allow you to move left, down, up, or right, never diagonally. Using IN we could check to see if both the 5 and 6 keys are pressed to enable movement both left and down, ie. diagonally towards the bottom left of the screen so that movement control could be more like that of a joystick. Try this program to draw a line going up and right from the bottom left corner towards the top right

```
10 LET X=0
20 LET Y=0
30 PLOT X,Y
40 LET A=IN 61438
50 LET X=X+(A=251 OR A=243)
60 LET Y=Y+(A=247 OR A=243)
70 GO TO 30
```

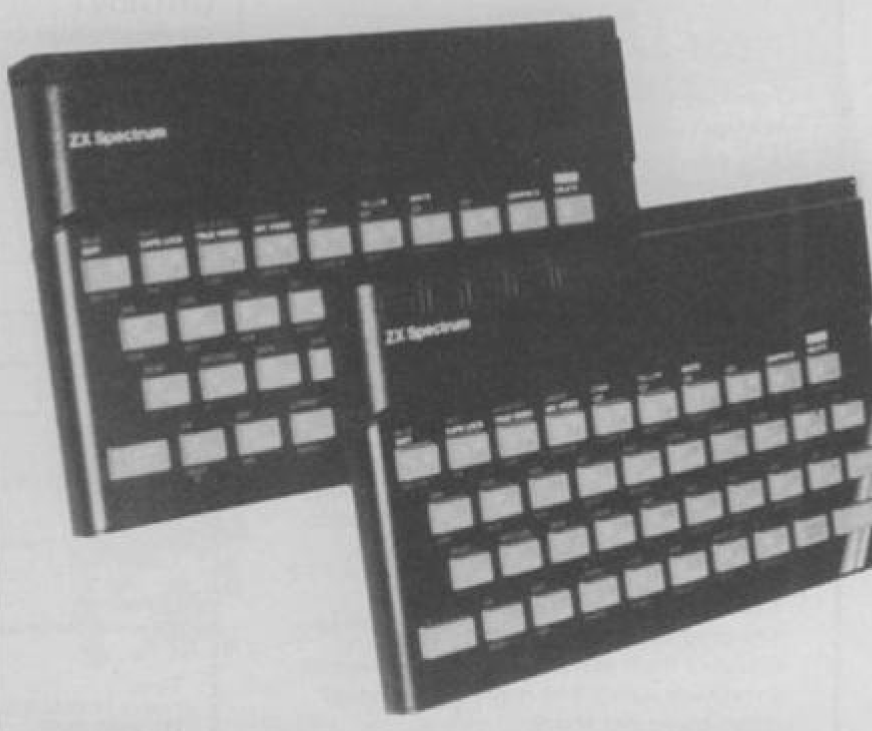
corner, rather like a graph. The controls are 7 to move up, 8 to move right, and press both 7 and 8 to move diagonally up and right. This would not have been so easy if we had used INKEY$ since we would not have been able to check if both the 7 and the 8 key were pressed.

Whilst on the subject of using the cursor keys 5,6,7,8 to control screen movements, wouldn't it be nice if this could be made easier to use? The reasons they are commonly used for this purpose are that they have direction arrows marked near them on the keyboard and they are easy to read with INKEY$ to control variable values (you may be familiar with, say, LET X = X + (INKEY$ = "8") − (INKEY$ = "5")) The snag is that these keys are so close together that it requires some pretty nimble finger action for fast, accurate control. The system to be described allows the use of the entire 40-key keyboard to control movement so that you don't have to worry so much about your finger being on the exact key required. The keyboard will be split into 4 (from the point of view of the program, not sawing it apart!) parts, each a block of 10 keys like this controlling movement in the directions shown.

1			UP		Ø
Q	LEFT	T	Y	RIGHT	P
A	LEFT	G	H	RIGHT	ENTER
CAPS SHIFT			DOWN		SPACE

So pressing any of the keys on the top row of the keyboard causes movement upwards; pressing any of the keys on the bottom row of the keyboard causes movement downwards. Pressing keys on the left half of the middle rows of the keyboard makes you move left and pressing keys on the right half of the middle rows of the keyboard makes you move right. Pressing different groups of keys have a combined effect, eg. if you pressed the 3 key and the W key you would move diagonally up and left. The program to demonstrate the routine is a very simple sketcher program which draws in the direction you're "steering" it. If you're not pressing any keys you stay still, as you'd expect. Do not expect this to be the best sketcher ever — it crashes if you go off the edge of the screen. Refer back to the diagram showing the I/O ports associated with the keyboard when examining lines 30 and 40 which do all the keyboard scanning.

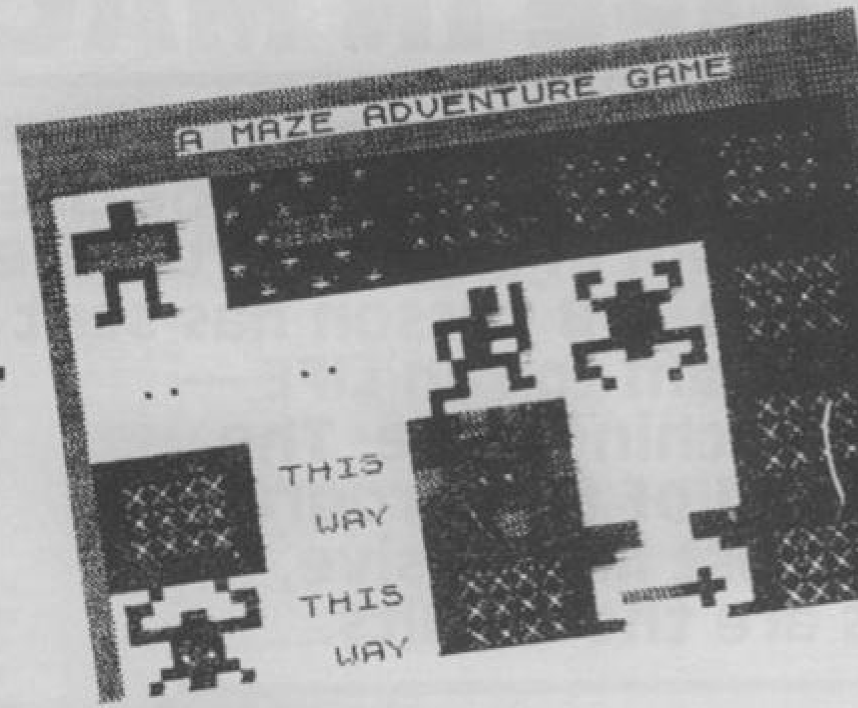
```
10 LET X=120
20 LET Y=90
30 LET X=X+(IN 49150<>255 OR I
N 57342<>255)-(IN 64510<>255 OR
IN 65022<>255)
40 LET Y=Y-(IN 65278<>255 OR I
N 32766<>255)+(IN 63486<>255 OR
IN 61438<>255)
50 PLOT X,Y
60 GO TO 30
```

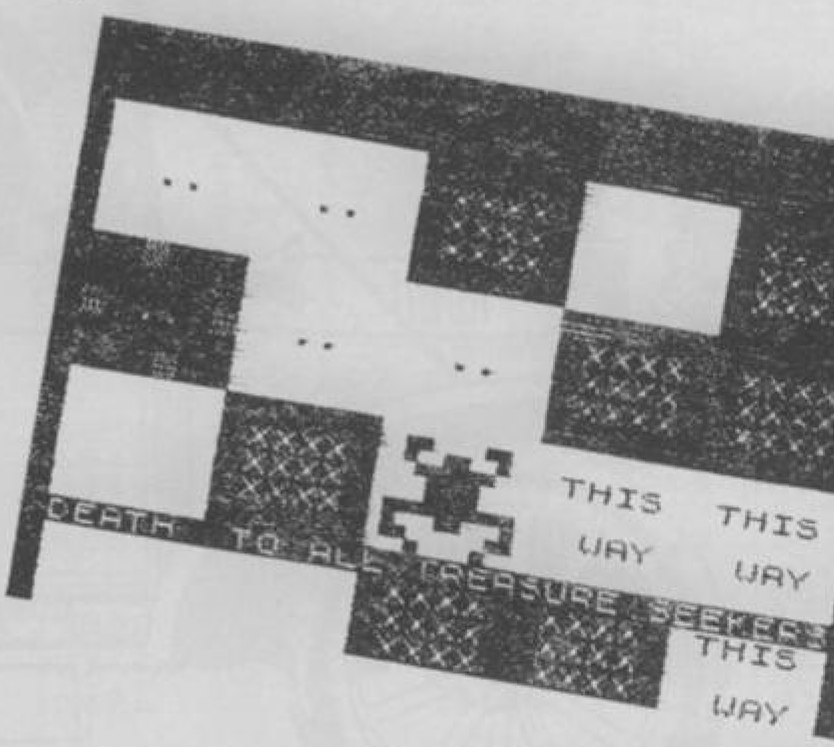


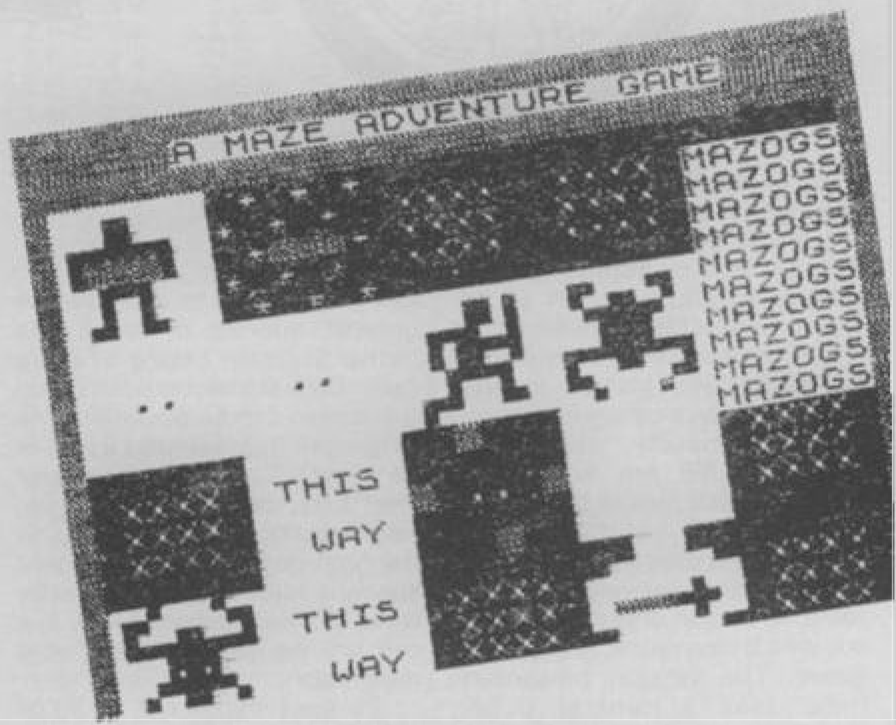
For the praise of Mazog

Reviewer Nick Pearce says he remembered seeing something rather blurred on a television screen on the Bug-Byte stand at a recent computer show — and it didn't look very exciting, so he passed it by. Now that he's had a good look at MAZOGS, Nick realises how wrong his first impression was.

Mazogs in action:







It's a pity I didn't take the time to study MAZOGS at that show. Since that first glimpse, I've seen mentions of the program everywhere, and now that I've played it I know what a marvellous game it is.

A new concept

MAZOGS is really a great new concept in computer games for the Sinclair, and in my opinion ranks alongside other such superb programs as J K Greye's MONSTER MAZE and Psion's FLIGHT SIMULATION. It really does show what some imagination, and a fair bit of hard work, can come up with for the trusty old ZX81.

Innovation, perhaps, is the key word. The majority of software houses are very clever at writing ZX81 adaptations of traditional games, or producing ZX81 versions of programs that were originally devised for its more expensive relations, but they rarely demonstrate real innovation. This, of course, is not to denigrate the very good and indeed necessary work that continues to be done by program authors. But it is software like MAZOGS that is going to help change the attitude of that quite large body of people who see the ZX81 as just a toy, albeit a quite clever toy.

MAZOGS is just the sort of program to impress your friends, and maybe convert a few sceptics into the bargain.

I will not explain the game in detail, as that might take away some of the fun. However, to whet your apetite, a *very* large and complex maze is involved. There are prisoners locked in the maze whose eyes blink, and who can guide you. There is treasure to find, swords for your protection, and — of course — battles to be fought with those nasty creatures, the Mazogs.

The instructions lack clarity, but it doesn't really matter. You'll have great fun learning the game by trying to play it.

Becoming an addict

The game should perhaps carry a warning. For once, the description 'addictive' is an accurate one, and you will not feel much like relinquishing the television set to such things as BBC and ITV after you've had a taste of MZG. User-friendly aspects have not been overlooked in the program, and there is none of that frustrating blank screen wait while the computer is 'thinking' or setting up the game.

All in all, my conclusion must be that this is one of the tiny minority of programs which every ZX81 owner should possess, even if only to amaze those who are sceptical of the ZX81's computing power.

MAZOGS, for the 16K ZX81, is available from Bug-Byte, 100 The Albany, Old Hall Street, Liverpool 3, for £10.00.

LIFE IN MACHINE CODE

In the last issue of ZX COMPUTING we discussed John Conway's game of LIFE, and gave a sample program in BASIC to run the LIFE process. From Epsom, K B Cusson has sent us a fine program to run LIFE — in machine code. The speed of this program is most impressive, as are the results.



Most computer enthusiasts will have either played or heard of the game of LIFE. Its one of those classic games which refuses to die out. Part of its lasting appeal lies in the fact that it is a creative game, as opposed to destructive games in the Space Invaders category.

Unfortunately, most versions of LIFE are written in BASIC, which makes them very slow-running, and detracts greatly from their interest. In an attempt to overcome this, a reduced screen area is often used, which also detracts from the game. The version presented here uses a central 0.5K machine code routine, and is capable of running at two generations per second using the whole screen.

The Rules

Life is supposed to simulate the growth, survival or death of a (rather stylised) colony of living cells. Each character position on the screen can be occupied by a single cell (represented by O) or be vacant. If a living cell has less than two, or more than three, neighbouring cells it will die in the next generation. If a vacant space is surrounded by exactly three live neighbours a new live cell will be created there next generation.

To start things off an initial colony pattern is loaded by hand, or alternatively a random loading routine can be used.

Facilities

1) A variable delay of 0 to 64 seconds can be set between generations to allow more time to study interesting developments.
2) The game can be halted indefinitely between generations then restarted when desired.
3) The game can be stopped and a colony pattern edited manually, then restarted.
4) Six strings are available for the storage of patterns. These are numbered 1 to 6 and can be loaded with a current pattern at any stage as desired. In addition string no. 6 is automatically loaded whenever the display is edited or an initial colony loaded.

Loading The Program

The machine code section is 435 bytes long and is stored in a REM statement at line 1. Before entering the code it is necessary to set up this REM statement with 435 dummy characters. The machine code loader program commencing at line 9000 should then be entered and started up using GOTO 9000. The program will prompt you with a request for a starting address, and you shold then input the opcodes in groups of five. At the end of each group a CHECKSUM will appear on the screen, and this should be com-

pared with the bracketed figures in the listing to ensure that no mistakes have been made. This procedure is used because machine code is tedious to enter and it is very easy to make mistakes which would almost certainly ruin the whole program. Also, as we know, the Sinclair printer is capable of producing the most atrocious print quality on occasions, making it difficult to tell the difference between similar characters. This particularly affects listings produced in hexadecimal, which are always liberally sprinkled with indistinguishable 'B's and '8's, and for this reason I have produced the listing in decimal. Should any ambiguity still occur in individual characters, the CHECKSUM can be used to resolve this, since it simply consists of the numerical sum of the preceeding five characters.

At the end of each group of codes the program will ask you "IS THE CHECKSUM CORRECT?". If you press Y the address of the next block of codes will be put up, which you should then enter. Pressing N erases the last block entered and reprints the start address.

When you have finished loading the REM statement, it is advisable to store it on tape, in case of the dreaded RAM pack wobbles. Whenever I am loading lengthy programs I save the intermediate result at 15 minute intervals to guard against this. There is nothing more frustrating than losing the last two hours labour because someone has clinked a teacup on the table at which you are working!

The rest of the program is in BASIC, lines 20 to 1600, and these are entered in the usual manner, except that the word LIFE in line 199 should have a normal letter E rather than an inverted one as its last character. Once entry of the BASIC is complete, it is necessary to dimension several strings which are used by the program, by entering the following direct commands:

DIM C$(884)
DIM U$(1)
DIM V$(1)
DIM W$(1)
DIM X$(1)
DIM Y$(1)
DIM Z$(1)

The completed program can then be saved on tape by entering the command GOTO 199. This will cause the program to save itself, then run.

Playing Life

Instructions are contained within the program, and these are reasonably self-explanatory, but a few points deserve emphasis. The game is menu driven, various option tables and instructions appearing as you proceed. Once the machine code section is entered, however, these disappear to be replaced by the colony pattern. Should you forget any of the control keys at this stage, the initial option page can always be recalled by pressing R.

Although the screen is two dimensional, it represents a spherical surface for the purposes of LIFE, ie the top line is considered to be adjacent to the bottom line, and the LHS and RHS columns are considered adjacent when deciding what the pattern of the next generation should be. This means that you are quite likely to see patterns disappearing off one side of the screen and reappearing at the other.

When loading start patterns by hand it is usually best to choose very simple ones, since they are easy to load and some of them will produce amazing results.

The Machine Code Loader:

```
MACHINE CODE LOADER

9000 PRINT "INPUT STARTING ADDRE
SS"
9010 DIM Y$(3)
9020 INPUT A
9030 FOR X=A TO 16948 STEP 5
9040 SCROLL
9050 LET CS=0
9060 PRINT AT 20,0;A;"    ";
9070 FOR Y=1 TO 5
9080 INPUT Y$
9090 PRINT Y$;"  ";
9100 POKE A,VAL Y$
9110 LET A=A+1
9120 LET CS=CS+VAL Y$
9130 NEXT Y
9140 PRINT "(";CS;")"
9150 PRINT AT 21,0;"IS THE CHECK
SUM CORRECT?(Y OR N)"
9160 LET B$=INKEY$
9170 IF B$<>"N" AND B$<>"Y" THEN
 GOTO 9160
9180 PRINT AT 21,0;"
                              "
9190 IF B$="N" THEN GOTO 9230
9200 NEXT X
9210 PRINT AT 21,0;"LOADING COMP
LETE"
9220 STOP
9230 PRINT AT 20,0;"
                              "
9240 LET A=A-5
9250 GOTO 9050
```

Codes to enter into Machine Code Loader:

16514	0	0	0	1	35	(36)
16519	0	235	9	235	6	(465)
16524	24	197	6	32	126	(385)
16529	18	35	19	16	250	(338)
16534	19	19	35	193	16	(282)
16539	241	43	43	1	51	(379)
16544	3	235	167	237	66	(708)
16549	235	6	32	126	18	(417)
16554	43	27	16	250	35	(371)
16559	1	33	0	235	9	(278)
16564	235	126	18	1	43	(428)
16569	3	235	9	235	6	(488)
16574	24	197	126	18	1	(366)
16579	33	0	167	237	66	(503)
16584	3	235	237	66	235	(776)
16589	193	16	239	1	33	(482)
16594	0	9	1	82	3	(95)
16599	235	9	235	126	18	(623)
16604	1	32	0	235	167	(435)
16609	237	66	235	65	126	(729)
16614	18	19	35	16	250	(338)
16619	43	1	33	0	235	(312)
16624	167	237	55	235	125	(831)
16629	18	1	48	3	167	(237)
16634	235	237	66	235	6	(779)
16639	24	197	126	18	1	(366)
16644	33	0	9	3	235	(280)
16649	9	235	193	16	242	(695)
16654	1	33	0	167	237	(438)
16659	66	1	82	3	235	(387)
16664	237	66	235	126	16	(682)
16669	201	235	1	35	0	(472)
16674	9	6	24	197	6	(242)
16679	32	14	0	213	17	(276)
16684	33	0	167	237	82	(519)
16689	126	254	52	32	1	(465)
16694	12	43	126	254	52	(487)
16699	32	1	12	43	126	(214)
16704	254	52	32	1	12	(351)
16709	19	25	126	254	52	(476)
16714	32	1	12	25	126	(196)
16719	254	52	32	1	12	(351)
16724	35	126	254	52	32	(499)
16729	1	12	35	126	254	(428)
16734	52	32	1	12	167	(264)
16739	237	82	126	254	52	(751)
16744	32	1	12	209	121	(375)
16749	254	2	40	11	254	(561)
16754	3	32	5	62	52	(154)
16759	16	24	2	175	18	(237)
16764	19	15	169	19	35	(258)
16769	35	193	16	160	201	(605)
16774	197	1	100	0	197	(495)
16779	6	30	0	0	0	(36)
16784	0	16	251	193	11	(471)
16789	121	176	32	241	193	(763)
16794	197	205	187	2	193	(784)
16799	124	165	254	255	32	(830)
16804	6	11	121	176	32	(346)
16809	220	201	205	167	2	(815)

```
16814   124 165 254 255 40   (836)
16819   247 68  77  33  254  (679)
16824   239 167 237 66  200  (909)
16829   33  251 247 167 237  (935)
16834   66  40  29  33  251  (419)

16839   223 167 237 66  40   (733)
16844   14  33  251 239 167  (704)
16849   237 66  32  214 175  (724)
16854   50  130 64  193 201  (638)
16859   62  1   50  130 64   (307)

16864   193 201 62  2   50   (508)
16869   130 64  193 201 235  (823)
16874   1   35  0   9   6    (51)
16879   24  197 6   32  126  (385)
16884   18  35  19  16  250  (338)

16889   35  35  19  193 16   (298)
16894   241 201 237 91  16   (786)
16899   64  33  6   0   25   (128)
16904   235 42  12  64  35   (388)
16909   201 205 0   66  205  (677)

16914   133 64  201 205 0    (603)
16919   66  205 133 64  205  (673)
16924   0   66  205 30  65   (366)
16929   237 75  131 64  3    (510)
16934   205 134 65  24  234  (662)

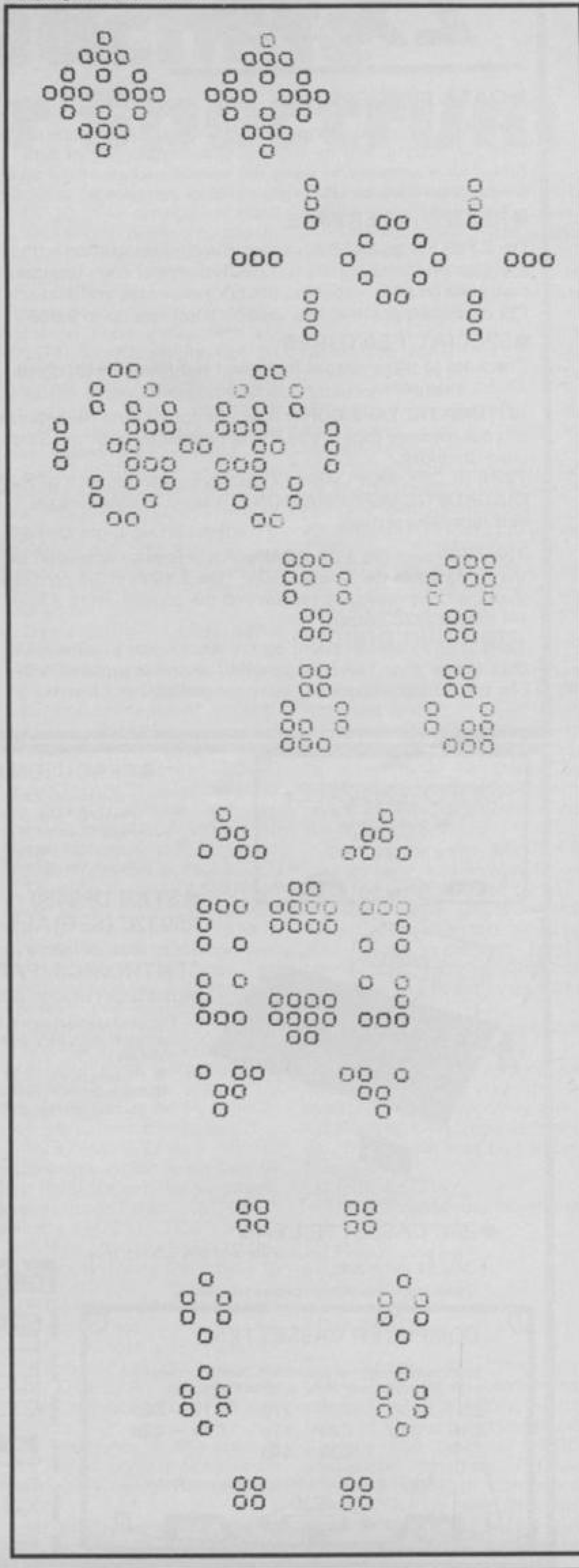
16939   205 0   66  205 233  (709)
16944   65  24  232 28  28   (377)
```

The program itself:

```
  40 REM STRING HANDLING SUBS.
  41 LET C$=U$
  42 RETURN
  43 LET C$=V$
  44 RETURN
  46 LET C$=W$
  47 RETURN
  49 LET C$=X$
  50 RETURN
  52 LET C$=Y$
  53 RETURN
  55 LET C$=Z$
  56 RETURN
  60 DIM U$(884)
  61 LET U$=C$
  62 RETURN
  63 DIM V$(884)
  64 LET V$=U$
  65 RETURN
  66 DIM W$(884)
  67 LET W$=C$
  68 RETURN
  69 DIM X$(884)
  70 LET X$=C$
  71 RETURN
  72 DIM Y$(884)
  73 LET Y$=C$
  74 RETURN
  75 DIM Z$(884)
  76 LET Z$=C$
  77 RETURN
  80 DIM U$(1)
  81 RETURN
  83 DIM V$(1)
  84 RETURN
  86 DIM W$(1)
  87 RETURN
  89 DIM X$(1)
  90 RETURN
  92 DIM Y$(1)
  93 RETURN
  95 DIM Z$(1)
  96 RETURN
 100 REM KEY EXECUTION SUB.
 102 PRINT AT L,C;" "
 104 IF (B$=CHR$ 114) OR (B$=CHR
$ 115) OR (B$=CHR$ 113) OR (B$=C
HR$ 112) THEN PRINT AT L,C;"O"
 106 IF (B$="5" OR B$=CHR$ 114)
AND C>0 THEN LET C=C-1
 108 IF (B$="8" OR B$=CHR$ 115)
AND C<31 THEN LET C=C+1
 110 IF (B$="6" OR B$=CHR$ 113)
AND L<21 THEN LET L=L+1
 112 IF (B$="7" OR B$=CHR$ 112)
AND L>0 THEN LET L=L-1
 114 IF B$="G" THEN GOTO 120
 116 PRINT AT L,C;"■"
 118 RETURN
 120 LET F=1
 122 RETURN
 150 REM DELAY ADJUST SUB.
 152 CLS
 154 PRINT AT 0,14;"LIFE"
 156 PRINT AT 2,0;"PLEASE INPUT
THE TIME DELAY      BETWEEN GENER
ATIONS. THIS CAN BEANY VALUE FRO
M 0 TO 64 SEC."
 158 INPUT T$
 160 IF T$="" THEN GOTO 186
 162 FOR C=1 TO LEN T$
 164 IF (T$(C)>"9" OR T$(C)<"0")
 AND T$(C)<>"." THEN GOTO 186
 166 NEXT C
 168 IF VAL T$<0 OR VAL T$>64 TH
EN GOTO 186
 170 LET T=10*VAL T$
 172 LET TM=INT (T/256)
 174 LET TL=INT (T-256*TM)
 176 POKE 16515,TL
 178 POKE 16516,TM
 180 PRINT AT 13,0;"TIME DELAY A
CCEPTED"
 182 PAUSE 25
 184 RETURN
 186 PRINT AT 13,0;"TRY AGAIN
   "
 188 GOTO 158
 199 SAVE "LIFE"
 200 CLS
 205 PRINT AT 0,14;"LIFE"
 210 PRINT AT 2,0;"YOUR INITIAL
OPTIONS ARE-"
 220 PRINT AT 4,0;"KEY T   SET IN
TER-GENERATION TIMEKEY A  RANDOM
LY LOAD DISPLAY     KEY M  MANUAL
LY LOAD DISPLAY     KEY S  LOAD A
 SAVED STRING"
 230 PRINT AT 10,0;"ONCE THE PRO
GRAM IS RUNNING-"
 240 PRINT AT 12,0;"PRESS ANY KE
Y TO HALT"
 250 PRINT AT 14,0;"KEY C  TO CO
NTINUE"
 260 PRINT AT 15,0,"KEY T  SET I
NTER-GENERATION TIMEKEY R  RETUR
N TO THIS PAGE"
 270 PRINT AT 17,0;"KEY E  TO ED
IT THE DISPLAY (USE       KEYS
AS FOR MANUAL LOAD)"
 300 LET B$=INKEY$
 310 IF B$="" THEN GOTO 300
 320 IF B$="A" THEN GOTO 700
 330 IF B$="T" THEN GOTO 400
 340 IF B$="S" THEN GOTO 800
 350 IF B$="M" THEN GOTO 410
 360 GOTO 300
 400 GOSUB 150
 405 GOTO 200
 410 REM MANUAL LOADING ROUTINE
 420 CLS
 430 PRINT AT 0,14;"LIFE"
 440 PRINT AT 2,0;"USE KEYS 5,6,
7 AND 8 TO MOVE THECURSOR AND TH
E SAME KEYS WITH   SHIFT TO LOAD
```

```
LIVE CELLS"
 450 PRINT AT 6,0;"AUTO-REPEAT I
S OBTAINED BY        HOLDING DOWN
THE KEYS"
 460 PRINT AT 10,0;"KEY M   TO LO
AD.WHEN COMPLETE      KEY G   TO GO
"
 470 LET B$=INKEY$
 480 IF B$<>"M" THEN GOTO 470
 490 CLS
 500 LET F=0
 510 LET L=11
 520 LET C=15
 530 PRINT AT L,C;"■"
 540 LET B$=INKEY$
 550 IF B$="" THEN GOTO 540
 560 GOSUB 100
 570 IF F=1 THEN GOTO 1510
 600 IF INKEY$<>B$ THEN GOTO 540
 610 GOSUB 100
 630 GOTO 600
 700 REM RANDOM LOADING ROUTINE"
 705 CLS
 710 RAND
 720 FOR L=0 TO 21
 730 FOR C=0 TO 31
 740 IF RND>0.8 THEN PRINT AT L,
C;"O"
 750 NEXT C
 760 NEXT L
 770 GOTO 1510
 800 REM STRING HANDLING ROUTINE
 810 CLS
 820 PRINT AT 0,14;"LIFE"
 830 PRINT AT 2,0;"KEY S   TO LOA
D FROM A STRING"
 840 PRINT AT 4,0;"KEY N   TO LOA
D TO A STRING"
 850 PRINT AT 6,0;"KEY E   TO ERA
ZE A STRING"
 860 PRINT AT 8,0;"KEY R   TO RET
URN TO INITIAL PAGE"
 870 PRINT AT 10,0,"KEY G   TO GO
"
 880 LET B$=INKEY$
 890 IF B$="" OR (B$<>"S" AND B$
<>"N" AND B$<>"E" AND B$<>"R" AN
D B$<>"G") THEN GOTO 880
 900 IF B$="S" THEN LET X=0
 910 IF B$="N" THEN LET X=20
 920 IF B$="E" THEN LET X=40
 930 IF B$="R" THEN GOTO 200
 940 IF B$="G" THEN GOTO 1580
 950 PRINT AT 13,0;"KEY IN STRIN
G NO. (1 TO 6)"
 960 LET B$=INKEY$
 970 IF B$="" OR B$>"6" OR B$<"0
" THEN GOTO 960
 980 IF B$="1" THEN LET Y=0
 990 IF B$="2" THEN LET Y=3
1000 IF B$="3" THEN LET Y=6
1010 IF B$="4" THEN LET Y=9
1020 IF B$="5" THEN LET Y=12
1030 IF B$="6" THEN LET Y=15
1040 GOSUB (40+X+Y)
1050 IF X=0 OR X=20 THEN PRINT A
T 13,0;"STRING LOADED OK
"
1060 IF X=40 THEN PRINT AT 13,0;
"STRING ERAZED OK
1070 GOTO 880
1500 REM USR HANDLING ROUTINE
1510 RAND USR 16910
1520 DIM Z$(884)
1530 LET Z$=C$
1540 RAND USR 16917
1550 IF PEEK 16514=0 THEN GOTO 2
00
1560 IF PEEK 16514=2 THEN GOTO 5
00
1570 GOSUB 150
1580 CLS
1590 RAND USR 16939
1600 GOTO 1550
```

The program in operation:



Mastering machine code on your Spectrum — part 2

In the last issue of ZX Computing, Toni Baker, author of 'Mastering Machine Code on Your ZX81', opened up the topic of machine code on the Spectrum. Here, she continues the discussion, concluding her venture into the realms of registers, addresses and other mysteries.

In this issue, I'll attempt to take your level of machine code knowledge up to a point where at least you know what it is and how to make programs out of it. In the next article I shall leave the *mechanics* of programming alone, and start to concentrate on the *ideas* behind sticking instructions together in the right order without getting lost.

Machine Code

The brain of the ZX-Spectrum is a little black chip called a Z80A — this is the device which carries out machine code programs. Contrary to popular belief, NO part of the Spectrum will carry out BASIC programs. There is a chip (called the ROM) which effectively "is" Spectrum BASIC, but the ROM itself is not a brain — the ROM is a very big machine code program — and no more.

To be a little more precise, the Z80A chip mentioned above is actually a brain without a memory, and the memory of the Spectrum has to come on separate chips. Let's take a closer look at memory, and how it affects machine code.

Using memory

There are many ways to think of memory. Some people relate it to numbers written on pieces of paper and stored in matchboxes or jam-jars. A more logical approach assumes a knowledge of programming — at least in BASIC — and so this is the approach I shall use.

You are familiar, I take it, with the BASIC concept of an array. Imagine then the effect of the statement DIM M(100). This then gives you a series of different variables called M(1), M(2), M(3), and so on up to M(100). Machine code memory is organised a little like this. A dimension statement is not needed (and does not exist) because the memory arrangement is already set up in advance, but the set up is as follows. Memory is a series of *one-byte* variables, (that is to say variables which can each store any number between 00 and FF) which are called (0000), (0001), (0002), and so on up to (FFFF). Notice I've written them in brackets because they are like elements of an array, but without any preceeding letter — this is because the array has no name — it doesn't need one because there is only one such array in existance in machine code.

Because memory is set up in advance, much of it is already used by the computer. This has absolutely nothing to do with the Z80A, it is simply a result of the way that the memory is wired up. (This makes it Sinclair's fault). It is used as follows.

0000-3FFF are already filled with various numbers, and in fact form a machine code program. You cannot change any of these addresses even if you want to — for instance the machine code instruction LD (02D4),A will simply not be carried out (although it will take the same amount of time to execute as LD (42D4),A which *would* work). These addresses are called ROM.

4000-5CB6 are addresses which all have a fixed purpose. For instance, (4000) stores the first eight pixels on the TV screen, and (5C8D) stores current colours (as used by PRINT statements in BASIC), and so on.

Depending upon the context, something like (ABCD) can mean one of two different things. Either

(i) The contents of address ABCD (exactly as you'd expect).

or (ii) It can also mean (ABCD) + 100*(ABCE) define as in (i). To make this clear, suppose that address ABCD stores the byte F3 and that address ABCE stores the byte 4A

Then by writing (ABCD) we could mean *either* the number F3, *or* the number 4AF3. Exactly which of the two forms we mean can be deduced fairly easily from the context. For instance 'LD A,(ABCD)' must mean 'LD A, F3' since the A register can only hold one byte, whereas 'LD HL,(ABCD)' must mean 'LD HL,4AF3' since the HL register pair must hold two bytes.

5CB7 to (5C65)-1 are all addresses which are used by the ROM for varying purposes. Note here that (5C65)-1 means (5C65) + 100*(5C66) − 1. The absolute value of this address will vary from time to time.

(5C65) to SP-1 are spare addresses not used by the ROM. You can use these for any purpose you choose — eg 'LD (7000),A' is generally quite sensible. Note though that:

(i) the bytes at the start of this region will are likely to be continually overwritten by the ROM.

(ii) the bytes at the end of this region will definitely be overwritten by something called an 'interrupt routine' which I shall talk about in the next article.

SP is a two-byte machine code register which I shall dwell on a bit more further on.

SP to (5CB2) is something called the MACHINE STACK which I shall explain in greater detail further on.

(5CB2) + 1 to (5C7B) − 1 is spare and can be used in machine code.

(5C7B) to (5CB4) − 1 are used by the ROM to store each of the user defined graphics.

(5CB4) to 7FFF if you have 16K

or FFFF if you have 48K are all spare addresses and can be used by us.

Storing numbers

If you have only 16K then addresses 8000 to FFFF do not exist.

Enough of all that boring drivel. Let's turn our attentions to something constructive. Firstly, here are all the different ways of storing numbers on the Spectrum.

Instructions like LD C,42 you already know about, so let's see what else we can do. I'll use some abbreviations here: the small letters m, n, p, and q to represent single bytes, and combinations mn and pq to represent byte pairs.

LD A,(pq) Only the A register may be used in this way.

LD (pq),A The reverse operation. Again only the A register may be used like this.

LD BC,(pq) Only register pairs BC, DE and HL may be used in this way.

LD (pq), BCDitto.

The following combinations can NOT be made:

LD (qp),m

LD (pq),mn

LD (pq),(mn)

The stack

The Stack, or "machine stack" as it is sometimes referred to, is a highly useful piece of memory organisation. It does not have a BASIC equivalent. Imagine, if you would, a stack of cardboard boxes. We may only alter this stack in two ways — we can place more boxes on top, or

we can remove boxes from the top. What we cannot do is to insert boxes into the middle, or to remove boxes from the middle. The machine stack is very similar to this idea, except that it uses numbers instead of cardboard boxes, for although cardboard boxes would probably be more fun, they wouldn't fit in the case of the Spectrum. It consists of a sequence of numbers.

You can place new numbers at the end of the list, and you can remove numbers from the list. Each number in the list is two bytes long.

To spot the usefulness of the stack it is best to show by example. This program will exchange the values of register pairs BC and DE — notice how it works:

PUSH BC Stack the number currently held in BC.
PUSH DE Stack the number currently held in DE.
POP DE Remove the number at the top of the stack and assign to BC.
POP DE Remove the number at the top of the stack and assign to DE.

The words "push" and "pop" are a delightful pair of words — almost as much fun as PEEK and POKE in fact. PUSH means append a new number to the list, and POP means remove the most recently PUSHed number from the list.

The actual location of the stack is somewhere very high in memory. The computer keeps track of things with a machine code register called SP (Stack Pointer). SP is a two-byte register, just like BC or HL, however, unlike these two it cannot be separated into its two constituent bytes. Unfortunately, for some reason as yet unknown to me, the instruction LD HL, SP does not exist, so if you ever want to know the exact value of SP you need two separate instructions: LD HL,0000/ADD HL, SP. What does SP do? I hear you ask! It simply stores the address of the item at the top of the stack — this register is updated automatically every time either PUSH or POP is used.

Arithmetic

The simplest arithmetic instructions are INC (increment) and DEC (decrement). INC D means increase D by one, DEC HL means decrease HL by one, and so on. You should notice however the following phenomena: if D contains the byte FF then INC D will change the value of D not to 100 (for it cannot contain a value that large) but to 00; similarly, if HL contains 0000 then DEC HL will "decrease" it to FFFF.

What we *can't* do, although I'm sure we'd all like to, is to be able to say LD A,B + C or LD BC,3 * HL + 1, almost as we do in BASIC. Unfortunately, such luxuries are beyond us, and 'LD' can only ever transfer the value of a register, a constant, or the contents of an address. Arithmetic is restricted to a few basic specialised instructions. In order to understand arithmetic we first need to introduce a new "register" called the CARRY FLAG.

The carry flag

A FLAG is very similar to a REGISTER except that where a register may hold any value between 00 and FF, a flag may only hold the number zero or the number one. A very, very important flag — as far as arithmetic is concerned — is the CARRY FLAG. I shall denote this flag by the letter K to avoid confusion with the C register. (Please note that this is not a standard convention, and that to my knowledge nobody uses this representation except me).

The need for the carry flag is made apparent by additions like 93 + E8. The correct answer is of course 17B, however, no single register on its own can store numbers that large. If A contained the number 93, and B contained the number E8, then "ADD A,B' (at first sight equivalent to the BASIC statement LET A = A + B) could never store the correct answer in the A register alone. The CARRY comes in as follows. I shall write KA to mean the "register pair" formed by K (the carry flag) and A (the A register).

If the result of such a subtraction turns out to be negative you should add 200 to KA, or 20000 to KHL, as the case may be. In other words, "LET KA = 04-07" means "LET KA = 200 + 04 − 07" (or 1FD) rather than simply minus three.

Some other instructions you should know are:

```
AND A meaning LET K = 0
SCF meaning LET K = 1
CCF meaning LET K = 1 - K
```

```
Some quite useful things to know are:

SBC A,A means IF K = 0 THEN LET A = 00
              IF K = 1 THEN LET A = FF
AND A/SBC HL,BC means LET KHL = HL - BC
```



Machine Code

Multiplication and division instructions do *not* exist in machine code, but we shall see how we may get around this in a later article. In the next part of this series I shall conclude my explanation of all machine code instructions, and begin to discuss how they may be usefully put together into PROGRAMS. In the meantime, here are four tables which between them tell you the hexadecimal codes for all of the machine code instructions, even the ones you don't know about. The tables are reproduced from my book MASTERING MACHINE CODE ON YOUR ZX81 — please keep them even though you may not understand them all as yet, for they will not be repeated.

```
ADD A,B means LET KA = A + B
ADC A,B means LET KA = A + B + K
ADD HL, BC means LET KHL = HL + BC
ADC HL,BC means LET KHL = HL + BC + K

similarly
SUB A,B means LET KA = A - B
SBC A,B means LET KA = A - B - K
SUB HL,BC does not exist
SBC HL,BC means LET KHL = HL - BC - K
```

TABLE ONE

r	B	C	D	E	H	L	(HL)	A	(IX+d)	(IY+d)	n
ADD A,r	80	81	82	83	84	85	86	87	DD86dd	FD86dd	C6nn
ADC A,r	88	89	8A	8B	8C	8D	8E	8F	DD8Fdd	FD8Fdd	CEnn
AND r	A0	A1	A2	A3	A4	A5	A6	A7	DDA6dd	FDA6dd	E6nn
BIT 0,r	CB40	CB41	CB42	CB43	CB44	CB45	CB46	CB47	DDCBdd46	FDCBdd46	—
BIT 1,r	CB48	CB49	CB4A	CB4B	CB4C	CB4D	CB4E	CB4F	DDCBdd4E	FDCBdd4E	—
BIT 2,r	CB50	CB51	CB52	CB53	CB54	CB55	CB56	CB57	DDCBdd56	FDCBdd56	—
BIT 3,r	CB58	CB59	CB5A	CB5B	CB5C	CB5D	CB5E	CB5F	DDCBdd5E	FDCBdd5E	—
BIT 4,r	CB60	CB61	CB62	CB63	CB64	CB65	CB66	CB67	DDCBdd66	FDCBdd66	—
BIT 5,r	CB68	CB69	CB6A	CB6B	CB6C	CB6D	CB6E	CB6F	DDCBdd6E	FDCBdd6E	—
BIT 6,r	CB70	CB71	CB72	CB73	CB74	CB75	CB76	CB77	DDCBdd76	FDCBdd76	—
BIT 7,r	CB78	CB79	CB7A	CB7B	CB7C	CB7D	CB7E	CB7F	DDCBdd7E	FDCBdd7E	—
CP r	B8	B9	BA	BB	BC	BD	BE	BF	DDBEdd	FDBFdd	FEnn
DEC r	05	0D	15	1D	25	2D	35	3D	DD35dd	FD35dd	—
IN r,(C)	FT40	ED48	ED50	ED58	ED60	ED68	—	Ed78	—	—	—
INC r	04	0C	14	1C	24	2C	34	3C	DD34dd	FD34dd	—
LD B,r	40	41	42	43	44	45	46	47	DD46dd	FD46dd	06nn
LD C,r	48	49	4A	4B	4C	4D	4E	4F	DD4Edd	FD4Edd	0Enn
LD D,r	50	51	52	53	54	55	56	57	DD56dd	FD56dd	16nn
LD E,r	58	59	5A	5B	5C	5D	5E	5F	DD5Edd	FD5Edd	1Enn
LD H,r	60	61	62	63	64	65	66	67	DD66dd	FD66dd	26nn
LD L,r	68	69	6A	6B	6C	6D	6E	6F	DD6Edd	FD6Edd	2Enn
LD (HL),r	70	71	72	73	74	75	—	77	—	—	36nn
LD A,r	78	79	7A	7B	7C	7D	7E	7F	DD7Edd	FD7Edd	3Enn
LD (IX+d),r	DD70 dd	DD71 dd	DD72 dd	DD73 dd	DD74 dd	DD75 dd	—	DD77 dd	—	—	DD36 ddnn
LD (IY+d),r	Fd70 dd	FD71 dd	FD72 dd	FD73 dd	FD74 dd	FD75 dd	—	FD77 dd	—	—	FD36 ddnn
OR r	B0	B1	B2	B3	B4	B5	B6	B7	DDB6dd	FDB6dd	F6nn
OUT (C),r	ED41	ED49	ED51	ED59	ED61	ED69	—	ED79	—	—	—
RL r											
RES 0,r	CB80	CB81	CB82	CB83	CB84	CB85	CB86	CB87	DDCBdd86	FDCBdd86	—
RES 1,r	CB88	CB89	CB8A	CB8B	CB8C	CB8D	CB8E	CB8F	DDCBdd8E	FDCBdd8E	—
RES 2,r	CB90	CB91	CB92	CB93	CB94	CB95	CB96	CB97	DDCBdd96	FDCBdd96	—
RES 3,r	CB98	CB99	CB9A	CB9B	CB9C	CB9D	CB9E	CB9F	DDCBdd9E	FDCBdd9E	—
RES 4,r	CBA0	CBA1	CBA2	CBA3	CBA4	CBA5	CBA6	CBA7	DDCBddA6	FDCBddA6	—
RES 5,r	CBA8	CBA9	CBAA	CBAB	CABC	CBAD	CBAE	CBAF	DDCBddAE	FDCBddAE	—
RFS 6,r	CBB0	CBB1	CBB2	CBB3	CBB4	CBB5	CBB6	CBB7	DDCBddB6	FDCBddB6	—
RFS 7,r	CBB8	CBB9	CBBA	CBBB	CBBC	CBBD	CBBE	CBBF	DDCBddBE	FDCBddBE	—
RLC r	CB00	CB01	CB02	CB03	CB04	CB05	CB06	CB07	DDCBdd06	FDBdd06	—
RRC r	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E	CB0F	DDCBdd0E	FDCBdd0E	—
RL r	CB10	CB11	CB12	CB13	CB14	CB15	CB16	CB17	DDCBdd16	FDCBdd16	—
RR r	CB18	CB19	CB1A	CB1B	CB1C	CB1D	CB1E	CB1F	DDCBdd1E	FDCBdd1E	—
SET 0,r	CBC0	CBC1	CBC2	CBC3	CBC4	CBC5	CBC6	CBC7	DDCBddC6	FDCBddC6	—
SET 1,r	CBC8	CBC9	CBCA	CBCB	CBCC	CBCD	CBCE	CBCF	DDCBddCE	FDCBddCE	—
SET 2,r	CBD0	CBD1	CBD2	CBD3	CBD4	CBD5	CBD6	CBD7	DDCBddD6	FDCBddD6	—
SET 3,r	CBD8	CBD9	CBDA	CBDB	CBDC	CBDD	CBDE	CBDF	DDCBddDe	FDCBddDE	—
SET 4,r	CBE0	CBE1	CBE2	CBE3	CBE4	CBE5	CBE6	CBE7	DDCBddE6	FDCBddE6	—
SET 5,r	CBE8	CBE9	CBEA	CBEB	CBEC	CBED	CBEE	CBEF	DDCBddEE	FDCBddEE	—
SET 6,r	CBF0	CBF1	CBF2	CBF3	CBF4	CBF5	CBF6	CBF7	DDCBddF6	FDCBddF6	—
SET 7,r	CBF8	CBF9	CBFA	CBFB	CBFC	CBFD	CBFE	CBFF	DDCBddFE	FDCBddFE	—
SUB A,r	90	91	92	93	94	95	96	97	DD96dd	FD96dd	D6nn
SBC A,r	98	99	9A	9B	9C	9D	9E	9F	DD9Edd	FD9Edd	DEnn
SLA r	CB20	CB21	CB22	CB23	CB24	CB25	CB26	CB27	DDCBdd26	FDCBdd26	—
SRA r	CB28	CB29	CB2A	CB2B	CB2C	CB2D	CB2E	CB2F	DDCBdd2E	FDCBdd2E	—
SRL r	CB38	CB39	CB3A	CB3B	CB3C	CB3D	CB3E	CB3F	DDCBdd3E	FDCBdd3E	—
XOR r	A8	A9	AA	AB	AC	AD	AE	AF	DDAEdd	FDAEdd	EEnn

TABLE TWO

s	BC	DE	HL	SP	IX	IY
ADC HL,s	ED4A	ED5A	ED6A	ED7A	—	—
ADD HL,s	09	19	29	39	—	—
ADD IX,s	DD09	DD19	—	DD39	DD29	—
ADD IY,s	FD09	FD19	—	FD39	—	FD29
DEC s	0B	1B	2B	3B	DD2B	FD2B
INC s	03	13	23	33	DD23	FD23
LD s,mn	01nnmm	11nnmm	21nnmm	31nnmm	DD21nnmm	FD21nnmm
LD s,(pq)	ED4Bqqpp	ED5Bqqpp	2Aqqpp	ED7Bqqpp	DD2Aqqpp	FD2Aqqpp
LD (pq),s	ED43qqpp	ED53qqpp	22qqpp	Ed73qqpp	DD22qqpp	FD22qqpp
POP s	C1	D1	E1	—	DDE1	FDE1
PUSH s	C5	D5	E5	—	DDE5	FDE5
SBC HL,s	ED42	ED52	ED62	ED72	—	—

PART 3 OF THIS ARTICLE WILL BE CARRIED OVER TO NEXT MONTH

MAKING SENSE OF BUBBLE SORTING



Many programmers have used bubble sort subroutines to get the contents of an array into order. Few of those who've used the routine could explain how it works. Mike Biddell, Tamworth, Staffordshire, takes on the task of making it all clear.

I wasn't sure how bubble sorts worked, so decided to try and fathom the whole thing out. I thought I'd experiment to investigate whether it was possible to devise programs which would make the process self-explanatory and thus make it clear what was happening. When I began doing this, I didn't really understand it myself. Now I think I do.

Although the bubble sort is not particularly fast or sophisticated, it is quite adequate for small arrays, and has the advantage of using up little memory.

Fast Bubble Sorting

To get an initial understanding of the bubble sort process, enter the program shown in Fig. 1. Examine and think about each program line as you do so. The program allows you to enter eight numbers of your choice in any order. The computer then enters the FAST mode and in a matter of seconds, prints them in ascending order. This gives you some idea of the capability of the bubble sort. An examination of the listing in Fig. 1, will reveal that you are asking the computer to carry out the following task, again and again:

"Take the first number at the top of the list and compare this in turn with each number (starting from the bottom of the list), until you find a smaller number; then swap these two numbers. (If no smaller number is found, move on to the second number from the top of the list and repeat the process)."

In this way, the small numbers 'bubble' to the top of the list and the heavier ones drop to the bottom. Use a pen and paper to write down eight numbers in a mixed order, then follow the instructions given to the computer, writing down the revised list after each swap. You will find that this simple process, does indeed sort the list into ascending order.

Slow Bubble Sorting

A clearer way to view the computer at work, following your programmed instructions, is to abandon the quest for speed and deliberately slow the computer down to a snail's pace and have it mark the two numbers it intends to swap before it does so. To obtain the slowest bubble sort in the history of computing, enter the program shown in Fig. 2 and run it. The computer prints two columns of numbers. The column on the left is the intial mixed array to be ordered and on the right is the column which changes as the sort progresses. The computer marks each pair to be swapped with black squares. It's fascinating to watch the computer at work in slow motion. (I sat mesmerised, running this program, for about twenty minutes). But the method of operation really sank into the grey matter by watching the process. When the sort is completed, the computer informs you and produces a display similar to that shown in Fig. 3.

At this point, I decided that one could improve the clarity of understanding by allowing the record of each swap to be retained on the screen (here you'll need the 16K RAM pack).

A few modifications to the program in Fig. 2 produced that shown in Fig. 4. This program obligingly records each swap made, across the screen, until sorting is complete. (See Fig. 5). This was a further step toward the full clarity of understanding.

An interesting empirical observation here, is that the number of swaps logged is always approximately equal to the number of elements in the array to be ordered. Is there a theoretical maximum? Perhaps the mathematicians amongst us could advise!

In Conclusion

Having whiled away a pleasant afternoon dabbling, I felt I had finally got the bubble sort 'sorted'. I hope these ramblings help you in the same way. I feel there's a lot more scope for slow computing and display to help us all understand what these little black boxes get up to. By the way, does anybody out there *really* understand the Shell-Metzner?

FIG. 1

```
  2 DIM A(8)
 10 FOR J=1 TO 8
 15 INPUT B
 16 LET A(J)=B
 17 PRINT A(J)
 20 NEXT J
 50 FOR J=1 TO 8
 52 FAST
 55 LET K=J+1
 60 FOR I=K TO 8
 65 LET S=K+8-I
 70 IF A(S)>A(J) OR A(S)=A(J) T
HEN GOTO 90
 75 LET M=A(S)
 80 LET A(S)=A(J)
 85 LET A(J)=M
 90 NEXT I
 95 NEXT J
 96 PRINT
 97 FOR J=1 TO 8
 98 PRINT A(J)
 99 NEXT J
```

FIG. 2

```
 2 DIM A(8)
 5 LET Y=0
10 FOR J=1 TO 8
15 LET A(J)=INT (RND*89+10)
20 NEXT J
21 LET X=0
22 FOR J=1 TO 8
23 FOR K=1 TO 20
24 NEXT K
25 PRINT AT X,Y;A(J)
28 LET X=X+2
30 NEXT J
40 LET Y=Y+3
50 FOR J=1 TO 8
55 LET K=J+1
60 FOR I=K TO 8
65 LET S=K+8-I
70 IF A(S)>A(J) OR A(S)=A(J) T
HEN GOTO 90
75 LET M=A(S)
80 LET A(S)=A(J)
85 LET A(J)=M
86 PRINT AT 2*S-2,Y-1;"█"
87 PRINT AT 2*J-2,Y-1;"█"
89 GOTO 21
90 NEXT I
95 NEXT J
96 PRINT "SORTED"
```

FIG. 3

```
94 25

33 26

55 29

29 33

49 35

35 49

25 55

26 94

SORTED
```

FIG. 4

```
 2 DIM A(8)
 5 LET Y=11
10 FOR J=1 TO 8
15 LET A(J)=INT (RND*89+10)
20 NEXT J
21 LET X=0
22 FOR J=1 TO 8
23 FOR K=1 TO 20
24 NEXT K
25 PRINT AT X,Y;" ";A(J)
28 LET X=X+2
30 NEXT J
40 LET Y=14
50 FOR J=1 TO 8
55 LET K=J+1
60 FOR I=K TO 8
65 LET S=K+8-I
70 IF A(S)>A(J) OR A(S)=A(J) T
HEN GOTO 90
75 LET M=A(S)
80 LET A(S)=A(J)
85 LET A(J)=M
86 PRINT AT 2*S-2,14;"█"
87 PRINT AT 2*J-2,14;"█"
89 GOTO 21
90 NEXT I
95 NEXT J
96 PRINT "SORTED"
```

FIG. 5

```
50█26█12  12  12  12  12

76  76  76█50█26  26  26

12  12█26  26█50  50  50

71  71  71  71  71  71  71

86  86  86  86  86█76  76

26█50  50█76  76█86  86

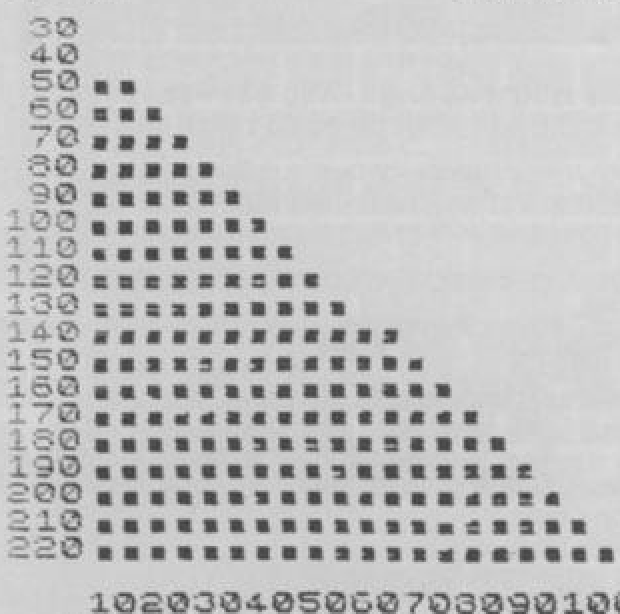
98  98  98  98  98  98█86

86  86  86  86  86  86█98
SORTED
```

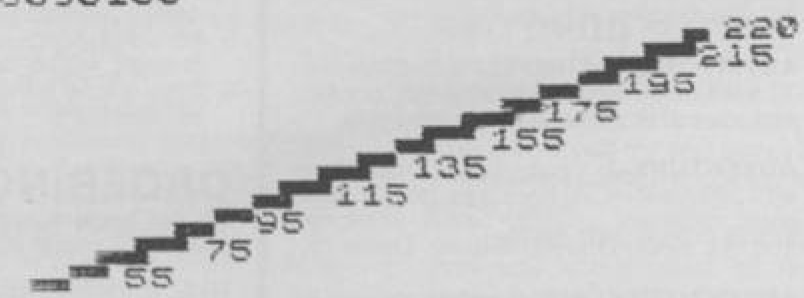
Conversion Time

Here are two programs for the ZX81 which convert temperatures from Fahrenheit to Centigrade. This is not, we know, particularly exciting, and there are many programs which do the task. However, we've included these two here because they approach it in quite a unique way.

The first one prints out a sort of chart which can be read to give an approximate conversion (read to the end of a line, then straight down). The second one creates a very interesting display, as it gradually draws up the 'steps', as you'll see when you run it. As an exercise, work out what the vertical scale should be, and add a few lines after line 70 to include this.




```
 10 REM CONVERSION GRAPH
 20 REM FAHR. TO CENT.
 30 FOR F=30 TO 220 STEP 10
 40 IF F<100 THEN PRINT " ";
 50 PRINT F;
 60 FOR C=1 TO 5*(F-32)/45
 70 PRINT "■";
 80 NEXT C
 90 PRINT
100 NEXT F
105 PRINT
110 PRINT TAB 3;"102030405060708090100"
```




```
10 REM CONVERSION GRAPH 11
20 REM FAHR. TO CENT.
30 FOR F=35 TO 220
50 IF F>40 THEN PRINT F;
60 PLOT 5*(F-32)/18,F/10
70 NEXT F
```

All in all, you're just another brick

Adam Waring, from Hull, has sent us two programs which he believes are the best he has ever written — THE WALL and SURROUND. In THE WALL you have to try and demolish the wall, before it demolishes you. SURROUND pits you against an intelligent ZX81, as you try to draw lines around each other.



The Wall

The wall is closing in on you. There's no escape. Can you destroy it before it destroys you? The answer, quite simply, is NO. The wall is endless. Your 'V' is moved rapidly from left to right at the top of the screen. The wall scrolls up towards you. Your only chance is to hurl yourself at the wall in the hope of dislodging some bricks.

The game is fast for a BASIC program. This is because characters are POKEd into the display file — much faster than PRINT AT, especially when dealing with numbers. Your score is POKEd on the bottom line of the screen, making use of the full 24 lines. It is necessary to use these lines as they are not moved up when SCROLL is used.

The speed of the wall moving can be made faster/slower by changing the length of the loop in line 250. You could even introduce another variable which makes the game faster as it goes along.

Variables used:-

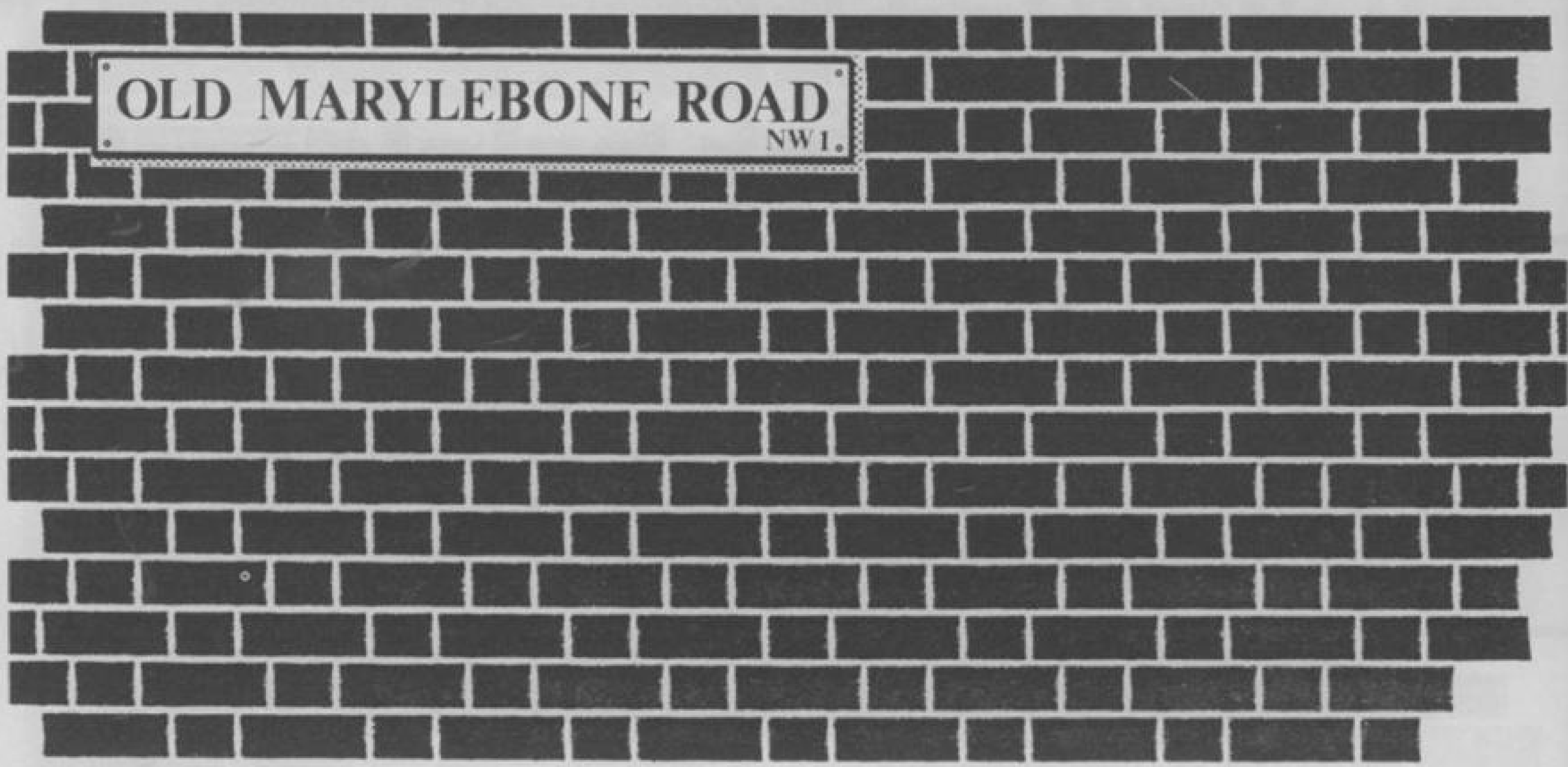
P = Display file for pokeing the 'V'.
W = Display file for pokeing your score.
S = Score.
H = High Score.
X = General purpose loops.
Y = Position for the 'V'.
C = Counter for number of bricks you have hit.
D = Number of bricks to be knocked out.
Z = Part of score being poked.
F = Vertical position for 'V' when being fired at the wall.
A$ = String array for printing "SPLAT" in big letters.
X$ = Score string to be poked on line 24.

Surround

The idea of this game is to trap your opponent, the ZX81, by making it bump into something. The ZX81 will try and trap you, using a rare blend of cunning and intelligence.

Despite this, you'll probably find you know what its next move is going to be after playing it a few times. Then you can get to work, and modify the computer's strategy by changing lines 540 to 570.

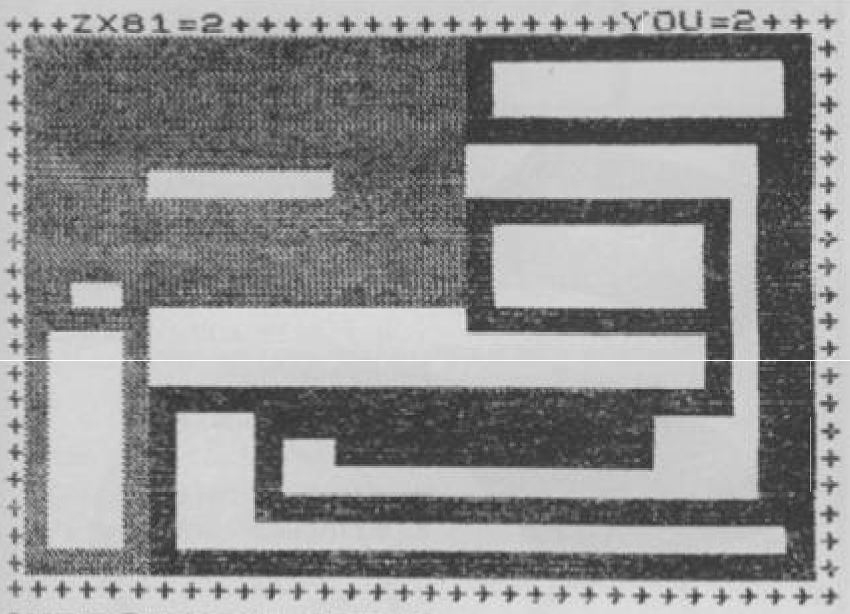
The first player to win five games wins the round.



WALL LISTING

```
  20 GOSUB 2000
  30 RAND
  40 LET P=PEEK 16396+PEEK 16397
*256+2
  50 LET W=P+765
  60 LET H=0
  70 DIM A$(7,32)
  80 GOSUB 900
  90 LET S=0
 110 FOR X=11 TO 21
 120 PRINT AT X,1;"[graphics]"
 130 NEXT X
 150 REM ** PRINT WALL **
 160 SCROLL
 170 PRINT "[graphics]"
 180 FOR X=0 TO 30
 190 IF PEEK (P+X)<>0 THEN GOTO
1000
 200 NEXT X
 240 REM ** MAIN LOOP **
 250 FOR X=1 TO 5
 260 FOR Y=0 TO 30
 280 POKE P+Y,59
 290 POKE P+Y-1,0
 300 IF INKEY$<>"" THEN GOTO 500
 310 NEXT Y
 320 POKE P+Y-1,0
 330 NEXT X
 340 GOTO 150
 500 REM ** FIRE **
 520 LET C=-1
 530 LET D=INT (RND*9)+1
 540 FOR F=0 TO 20 STEP 2
 550 IF PEEK (P+Y+F*33)<>0 THEN
LET C=C+1
 560 POKE P+Y+F*33,59
 570 IF PEEK (P+Y+(F+1)*33)<>0 T
HEN LET C=C+1
 580 POKE P+Y+(F+1)*33,0
 590 POKE P+Y+F*33,0
 600 IF C>=D THEN GOTO 620
 610 NEXT F
 620 LET S=S+C
 630 LET Z=S
 640 LET Z=Z/1000
 650 FOR F=0 TO 3
 660 POKE W+F,INT Z+156
 670 LET Z=Z-INT Z
 680 LET Z=Z*10
 690 NEXT F
 700 GOTO 330
 900 REM ** SCORE BOARD **
 920 LET X$="SCORE=0000 [graphics]
I-SCORE=0000"
 930 FOR X=1 TO 31
 940 POKE P+X+758,CODE X$(X)
 950 NEXT X
 960 RETURN
1000 REM ** SPLAT **
1010 LET A$(1)="[graphics]"
1020 LET A$(2)="[graphics]"
1030 LET A$(3)="[graphics]"
1040 LET A$(4)="[graphics]"
1050 LET A$(5)="[graphics]"
1060 LET A$(6)="[graphics]"
1070 LET A$(7)="[graphics]"
1100 FOR X=1 TO 7
1110 SCROLL
1120 PRINT A$(X)
1130 NEXT X
1140 FOR X=1 TO 14
1150 SCROLL
1160 PRINT "[graphics]"
1170 NEXT X
1180 PRINT AT 9,1;"YOU HAVE BEEN
 CRUSHED TO PIECES BY THE WALL.
   BITS OF YOU ARE ALL OVER THE
PLACE."
1190 PRINT AT 12,1;"YOU DEMOLISH
ED ";S;" BRICKS."
1200 PRINT AT 14,1;
1210 IF S<300 THEN PRINT "YOU AR
E PATHETIC.      HINT: TRY OPENIN
G YOUR EYES NEXT TIME."
1220 IF S>=300 AND S<500 THEN PR
INT "I SUGGEST YOU TAKE UP GOLF.
"
1230 IF S>=500 AND S<1000 THEN P
RINT "NOT BAD, BUT YOU COULD DO
WITH  SOME PRACTISE."
1240 IF S>=1000 AND S<3000 THEN
```

```
PRINT "YOUR SCORE IS AVERAGE."
1250 IF S>=3000 AND S<5000 THEN
PRINT "PRETTY GOOD.       YOU WONT
 DO SO WELL NEXT TIME."
1260 IF S>=5000 AND S<9000 THEN
PRINT "I DONT BELIVE MY EYES. HO
W CAN  SUCH A CRETIN DO SO GOOD?
"
1270 IF S>=9000 THEN PRINT "YOU
ARE THE BEST PLAYER I HAVE  BEAT
EN. YOU WILL BE REMEMBERED  FOR
YEARS TO COME."
1300 REM ** HI-SCORE **
1320 IF S<H THEN GOTO 1450
1330 LET H=S
1340 PRINT AT 17,8;"*CONGRATULAT
IONS*"
1350 PRINT " YOU HAVE OBTAINED T
HE HI-SCORE."
1360 LET Z=H
1370 LET Z=Z/1000
1380 FOR F=0 TO 3
1390 POKE P+F+786,INT Z+156
1400 LET Z=Z-INT Z
1410 LET Z=Z*10
1420 NEXT F
1450 PRINT AT 20,5;"ANOTHER GAME
? (Y OR N)"
1460 IF INKEY$="N" THEN STOP
1470 IF INKEY$<>"Y" THEN GOTO 14
60
1480 FOR X=0 TO 21
1490 PRINT AT X,1;"
                              "
1500 NEXT X
1510 GOTO 90
1600 REM ** LOAD AND GO **
1610 SAVE "WAL█"
1620 RUN
2000 REM ** INSTRUCTIONS **
2050 PRINT "THE WALL"
2060 PRINT "--------"
2070 PRINT
2080 PRINT "DEMOLISH THE WALL BE
FORE IT"
2090 PRINT "DEMOLISHES YOU."
2100 PRINT "YOUR ""V"" IS MOVED
RAPIDLY ACROSS"
2110 PRINT "THE SCREEN FROM LEFT
 TO RIGHT."
2120 PRINT "THE WALL MOVES SLOWL
Y TOWARDS"
2130 PRINT "YOU FROM THE BOTTOM
OF THE"
2140 PRINT "SCREEN. WHEN IT REAC
HES THE TOP"
2150 PRINT "LINE YOU CEASE TO EX
IST. YOUR"
2160 PRINT "SOLE DEFENCE IS TO H
URL YOURSELF"
2170 PRINT "AGAINST THE WALL IN
THE HOPE TO"
2180 PRINT "KNOCK SOME BRICKS OU
T. PRESS ANY"
2190 PRINT "KEY TO DO THIS."
2200 PRINT "PRESS ""S"" TO START
."
2210 IF INKEY$<>"S" THEN GOTO 22
10
2220 CLS
2230 RETURN
```



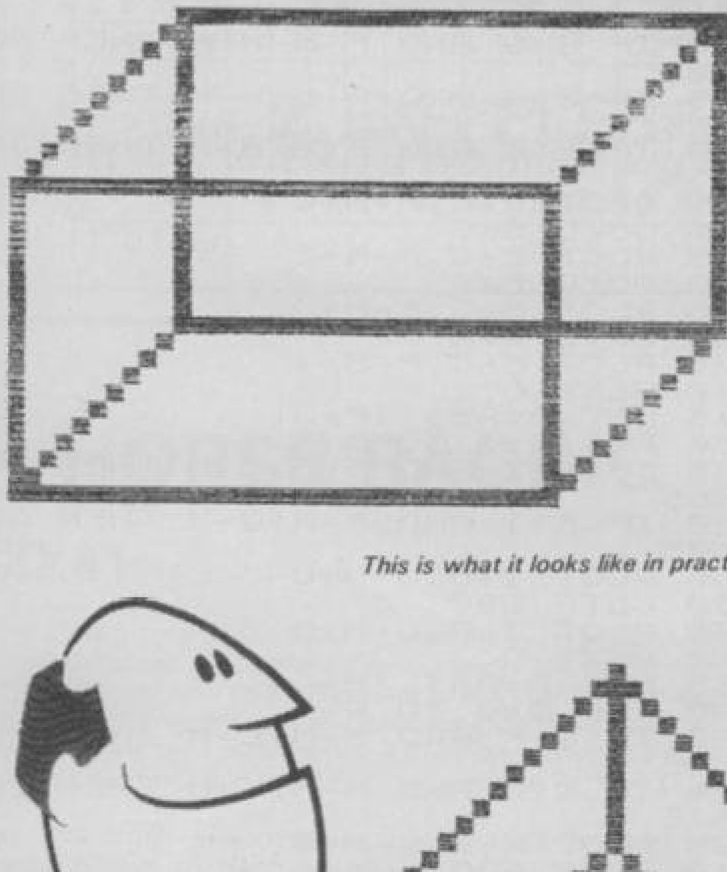
SURROUND LISTING

```
  10 CLS
  20 LET Z=0
  30 LET Y=Z
  50 LET A$="5"
  60 LET A=390
  70 LET P=PEEK 16396+PEEK 16397
*256+1
  80 LET B=334

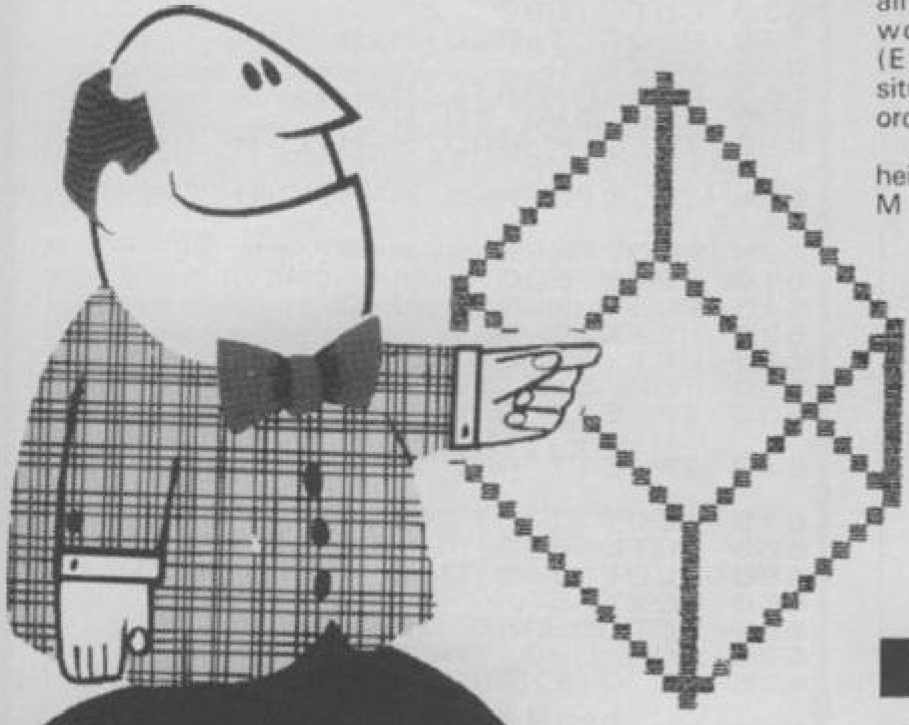
 120 FOR C=6 TO 20
 130 PRINT AT C,31;"++"
 140 NEXT C
 150 FOR C=0 TO 31
 160 PRINT AT 0,C;"+"
 170 PRINT AT 21,C;"+"
 180 NEXT C
 185 LET C=1
 190 PRINT AT 0,3;"ZX81=";Z;TAB
24;"YOU=";Y
 210 LET B=B+C
 220 IF PEEK (P+B)<>0 THEN GOSUB
 500
 250 LET B$=A$
 260 LET A$=INKEY$
 270 IF A$<"5" OR A$>"8" THEN LE
T A$=B$
 280 LET A=A+(A$="8")-(A$="5")+(
(A$="6")-(A$="7"))*33
 290 IF PEEK (P+B)<>0 THEN GOTO
1500
 300 POKE P+B,136
 310 IF PEEK (P+A)<>0 THEN GOTO
1000
 320 POKE P+A,128
 390 GOTO 200
 530 LET B=B-C
 540 LET C=33
 550 IF PEEK (B+P+1)=0 THEN LET
C=1
 560 IF PEEK (B+P-1)=0 THEN LET
C=-1
 570 IF PEEK (B+P-33)=0 THEN LET
 C=-33
 580 LET B=B+C
 590 RETURN
1050 CLS
1060 PRINT "THE ZX81 WINS YET AG
AIN"
1070 LET Z=Z+1
1075 PAUSE 4E4
1080 IF Z<5 THEN GOTO 40
1090 PRINT "ANOTHER ROUND TO ME"
1100 GOTO 1600
1550 CLS
1560 PRINT "YOU WON, WHAT A RARE
 OCCASION"
1570 LET Y=Y+1
1575 PAUSE 4E4
1580 IF Y<5 THEN GOTO 40
1590 PRINT "A ROUND TO YOU - AMA
ZING"
1600 PRINT "ZX81=";Z,"YOU=";Y
1610 PRINT "ANOTHER ROUND (Y OR
N)"
1620 LET A$=INKEY$
1630 IF A$="Y" THEN RUN
1640 IF A$<>"N" THEN GOTO 1620
```

THREE-DIMENSIONAL CUBES ON THE ZX81

Luc De Jaeger, from Slotendries, Belgium, has discovered a way to get representations of three-D cubes on the ZX81. He explains his method, which uses a 'selfmade television screen worksheet'.



This is what it looks like in practice:



Here's the way my program works. All you need is a 16K ZX81, a worksheet like the one I've shown here, and a little patience. Although it takes a while to type the program into your ZX81, the results seem pretty good to me.

How It Runs

Lines 5 to 65 ask for the four coordinates of the angular points of the upper surface of the cube. You only have to input the first co-ordinate of the first angular point (E;F), then press NEWLINE (of course), and then input the second coordinate, i.e. F . . . . . Mind the right order of the angular points!! Look first of all at the television screen worksheet to know where (E,F),(G,H),(I,J),(K,L) are situated, and don't change the order.

Line 70 asks you to give the height of the cube. Don't make M too high otherwise the cube will 'run out of the screen'. A height of about 10 to 25 is recommended. Of course, it all depends upon the position of the chosen coordinates of the angular points.

Lines 85 to 180 print the first 'horizontal' edge (see the second example) of the upper surface of the cube. In the first example the first line of the upper surface at the left will be printed.

Lines 185 to 280 print the second 'horizontal' edge.

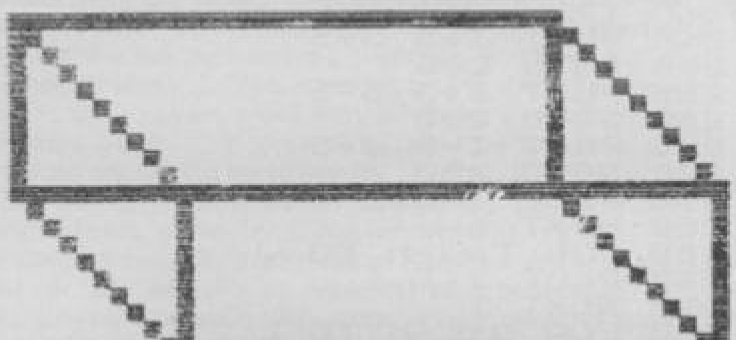
Lines 285 to 340 print the four vertical edges of the cube.

Lines 345 to 440 will print the first horizontal edge of the base (see example two) or the first line in example one.

Lines 445 to 540 do the same as lines 345 to 440. Then the second line will be printed.

Lines 545 to 700 print the four oblique lines of the cube.

As you can see the program is rather easy to understand because it's all BASIC.



PROGRAM LISTING

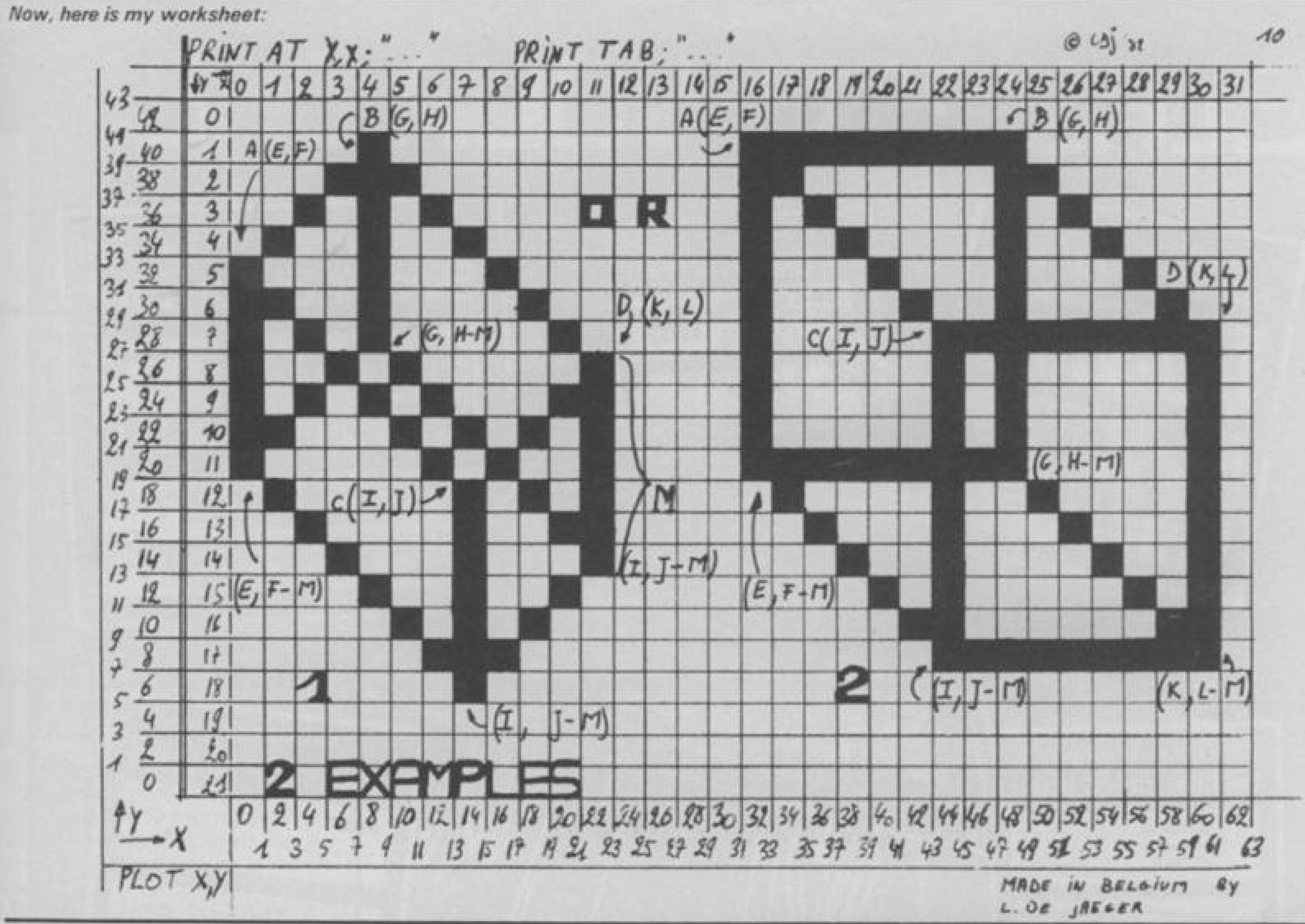
And here is the program listing:

```
1 REM **3-DIMENSIONAL CUBES**

3 REM **GENT - BELGIUM**
5 PRINT "INPUT ANGULAR POINTS
```

```
OF THE UPPER SURFACE"
  7 PRINT
 10 PRINT "INPUT ANGULAR POINT
A(E,F)"
 15 INPUT E
 20 INPUT F
 25 PRINT "INPUT ANGULAR POINT
B(G,H)"
 30 INPUT G
 35 INPUT H
 40 PRINT "INPUT ANGULAR POINT
C(I,J)"
 45 INPUT I
 50 INPUT J
 55 PRINT "INPUT ANGULAR POINT
D(K,L)"
 60 INPUT K
 65 INPUT L
 70 PRINT "WHAT IS THE HIGHT OF
 THE CUBE ?"
 75 INPUT M
 80 CLS
 85 IF F=H THEN GOTO 95
 90 IF F<>H THEN GOTO 115
 95 FOR N=E TO G
100 PLOT N,F
105 NEXT N
110 GOTO 185
115 LET O=ABS (F-H)
120 FOR P=0 TO O-1
125 IF E<G AND F>H THEN GOTO 15
5
130 IF E<G AND F<H THEN GOTO 14
5
135 IF E>G AND F<H THEN GOTO 16
5
140 IF E>G AND F>H THEN GOTO 17
5
145 PLOT E+P,F+P
150 GOTO 180
155 PLOT E+P,F-P
160 GOTO 180
165 PLOT E-P,F+P
170 GOTO 180
175 PLOT E-P,F-P
180 NEXT P
185 IF J=L THEN GOTO 195
190 IF J<>L THEN GOTO 215
195 FOR Q=I TO K
200 PLOT Q,J
205 NEXT Q
210 GOTO 285
215 LET R=ABS (J-L)
220 FOR S=0 TO R-1
225 IF I<K AND J>L THEN GOTO 25
5
230 IF I<K AND J<L THEN GOTO 24
5
235 IF I>K AND J<L THEN GOTO 26
5
240 IF I>K AND J>L THEN GOTO 27
5
245 PLOT I+S,J+S
250 GOTO 280
255 PLOT I+S,J-S
260 GOTO 280
265 PLOT I-S,J+S
270 GOTO 280
275 PLOT I-S,J-S
280 NEXT S
285 FOR T=J-M TO J
290 PLOT I,T
295 NEXT T
300 FOR U=L-M TO L
305 PLOT K,U
310 NEXT U
315 FOR V=F-M TO F
320 PLOT E,V
325 NEXT V
330 FOR W=H-M TO H
335 PLOT G,W
340 NEXT W
345 IF J-M=L-M THEN GOTO 355
350 IF J-M<>L-M THEN GOTO 375
355 FOR X=I TO K
360 PLOT X,J-M
365 NEXT X
370 GOTO 445
375 LET Y=ABS (J-L)
380 FOR Z=0 TO Y-1
385 IF I<K AND J-M>L-M THEN GOT
O 415
390 IF I<K AND J-M<L-M THEN GOT
O 405
395 IF I>K AND J-M<L-M THEN GOT
O 425
400 IF I>K AND J-M>L-M THEN GOT
O 435
405 PLOT I+Z,J-M+Z
410 GOTO 440
415 PLOT I+Z,J-M-Z
420 GOTO 440
425 PLOT I-Z,J-M+Z
430 GOTO 440
435 PLOT I-Z,J-M-Z
440 NEXT Z
445 IF F-M=H-M THEN GOTO 455
450 IF F-M<>H-M THEN GOTO 475
455 FOR A=E TO G
460 PLOT A,F-M
465 NEXT A
470 GOTO 545
475 LET B=ABS (F-H)
480 FOR C=0 TO B-1
485 IF E<G AND F-M>H-M THEN GOT
O 515
490 IF E<G AND F-M<H-M THEN GOT
O 505
495 IF E>G AND F-M<H-M THEN GOT
O 525
500 IF E>G AND F-M>H-M THEN GOT
O 535
505 PLOT E+C,F-M+C
510 GOTO 540
515 PLOT E+C,F-M-C
520 GOTO 540
525 PLOT E-C,F-M+C
530 GOTO 540
535 PLOT E-C,F-M-C
540 NEXT C
545 LET D=ABS (F-J)
550 FOR O=0 TO D-1
555 IF E>I AND F-M>J-M THEN GOT
O 565
560 IF E<I AND F-M>J-M THEN GOT
O 575
565 PLOT I+O,J-M+O
570 GOTO 580
575 PLOT I-O,J-M+O
580 NEXT O
585 LET P=ABS (H-L)
590 FOR Q=0 TO P-1
595 IF G>K AND H-M>L-M THEN GOT
O 605
600 IF G<K AND H-M>L-M THEN GOT
O 615
605 PLOT K+Q,L-M+Q
610 GOTO 620
615 PLOT K-Q,L-M+Q
620 NEXT Q
625 LET R=ABS (F-J)
630 FOR S=0 TO R-1
635 IF E>I AND F>J THEN GOTO 64
5
640 IF E<I AND F>J THEN GOTO 65
5
645 PLOT I+S,J+S
650 GOTO 660
655 PLOT I-S,J+S
660 NEXT S
665 LET T=ABS (H-L)
670 FOR U=0 TO T-1
675 IF G>K AND H>L THEN GOTO 68
5
680 IF G<K AND H>L THEN GOTO 69
5
685 PLOT K+U,L+U
690 GOTO 700
695 PLOT K-U,L+U
700 NEXT U
```

Now, here is my worksheet:



BLAST-OFF YOUR ZX81



And explore a galaxy of NEW software!

All these titles are available now from your local Bookstore or Computershop. In case of difficulty, send cost of tape plus 25p postage per tape with the coupon to: Software Masters Ltd., 30 Lincoln Road, Olton, Birmingham B27 6PA. Trade enquiries welcome!

SM2015	Breakout	16K	5.95
SM2011	Computacalc	16K	7.95
SM2022	Constellation	16K	8.00
SM2023	Football Manager	16K	7.95
SM2003	Labyrinth	16K	5.95
SM2025	Magnus	16K	9.95
SM2021	Mazogs	16K	10.00
SM2017	Mugsy	16K	5.95
SM2018	Murgatroyds	16K	5.95
SM2019	Murgatroyds Revenge	16K	5.95
SM2005	Nightmare Park/Music	16K	6.95
SM2027	P.E.P.	16K	5.95
SM2012	Personal Banking System	16K	9.95
SM2024	Pilot	16K	5.95
SM2020	Progmerge	16K	5.95
SM2026	Puckman	16K	5.95
SM2008	Space Intruders	16K	5.95
SM2014	Space Invaders/Space Rescue	16K	6.95
SM2013	Star Trek/3D Os & Xs	16K	6.95
SM2007	ZX 81 Chess	16K	6.90
SM2001	Bumper 7	1K	5.95
SM2016	1K Games Pack	1K	6.00
SM2002	1K Super Trio	1K	7.95

Prices include VAT

Please send me the following programs

Code	Title	Cost
		£

Postage:- ______

I enclose Postal Order/Cheque for Total £ ______

Name ______

Address ______

ZXC

Software Masters Ltd.
30 Lincoln Road, Olton, Birmingham B27 6PA, England. Telephone: 021-707 7544.

FANCY A DROP OF BUBBLY?

Paul Holmes has written two great versions of the program LEMONADE STAND. One fits within 1K, and the other version, which has many more features, demands extra RAM.

Firstly we'll look at the 1K version.

In this program, the object is to make as much money as you can, by selling lemonade on a lemonade stand. You are given a report (such as "RAINY" or "ROAD WORKS NEARBY"), and the price per cup the lemonade will cost you. Based on that information, you decide how many cups you will make for the day and at what price you will sell them. If you sell at too high a price, then people will be more reluctant to buy. You will have to judge from experience how many cups you will make, but you can't make more cups than you can afford with the cash available.

First you have to initialise the different reports and their characteristics, using the first program listed.

RUN this, and then enter the following information, pressing NEWLINE/ENTER after each item:

ROADWORKS NEARBY,10,40
BAD WEATHER,5,25
FAIR WEATHER,70,75
GOOD WEATHER,90,110
HEATWAVE,150,150
STORMY,5,5

Once you have done this, enter the second program, which will delete the first, but leave the data intact. If you save the program as it is, the data will be saved along with the program. However, you will lose the data if you press RUN or CLEAR, so start by entering GOTO 10. Once you have entered your values, the program will give you a sales report. There will be a brief pause, and a new day will dawn.



1K PROGRAM LISTING

```
  10 DIM A$(6,18)
  20 FOR Z=1 TO 6
  30 INPUT A$(Z)
  40 INPUT A
  50 INPUT B
  60 LET A$(Z,17)=CHR$ A
  70 LET A$(Z,18)=CHR$ B
  80 PRINT A$(Z)
  90 NEXT Z
```

```
  10 LET P=2
  20 LET M=1000
  30 LET L=INT (RND*6+1)
  35 LET D=INT (RND*6+1)
  40 PRINT "REPORT: ";A$(D)( TO
16)
  50 PRINT "LEMONADE: ";P;"C"
  60 PRINT "CASH: $";M/100
  70 PRINT "CUPS?"
  80 INPUT C
  90 IF C*P>M THEN GOTO 80
 100 PRINT "PRICE?"
 110 INPUT L
 120 LET N=RND*CODE (A$(D,17))+R
ND*CODE (A$(D,18))
 130 LET A=INT (N-(1/10-(L-P))*N
)
 140 IF A>C THEN LET A=C
 150 CLS
 160 PRINT A;" CUPS SOLD"
 170 PRINT "COST YOU: ";C*P
 180 PRINT "CASH TAKEN: ";A*L
 190 PRINT "PROFIT: ";L*A-C*P
 200 LET M=M+(A*L-C*P)
 210 FOR G=1 TO 200
 215 NEXT G
 220 CLS
 230 IF RND>.7 THEN LET P=P+1
 250 GOTO 30
```

Now here is the 2K version.

The extended memory version, as you can see, looks much more attractive, and even creates a picture of the lemonade stand. The words in inverse graphics are:

40 REPORT
50 LEMONADE . . PER CUP
60 MONEY
85 CUPS MADE FOR TODAY
105 TO BE SOLD AT . . PER CUP
145 *LEMONADE*
220 CLOSED
280 SALES REPORT
300 CUPS SOLD
310 MONEY IN THE TILL
320 WHOLESALERS BILL
325 RENT
340 PROFIT

2K PROGRAM LISTING

```
  10 LET P=2
  20 LET M=1000
  25 LET R=50
  30 LET D=INT (RND*6+1)
  35 PRINT "         LEMONADE S
TAND        "
  40 PRINT " REPORT        ";A$(D,
TO 16);"    "
  50 PRINT " LEMONADE      ";P;AT 2,
13;"P PER CUP        "
  60 PRINT " MONEY         £";M/100;T
AB 16;"              "
  70 PRINT AT 8,3;"HOW MANY CUPS
 TO BE MADE ?"
  75 INPUT C
  80 IF C*P>M THEN GOTO 75
  85 PRINT AT 4,0;" ";C;TAB 5;"
CUPS MADE FOR TODAY      "
  90 PRINT AT 8,3;"SALE PRICE ?
"
  95 INPUT L
 100 IF L>200 THEN GOTO 95
 105 PRINT AT 5,0;" TO BE SOLD A
T";L;TAB 18;"P PER CUP      "
 110 PRINT AT 8,3;"
"
 115 PRINT AT 6,0;"
"
 120 PRINT AT 8,3;"PRESS A KEY T
O OPEN STALL"
 125 IF INKEY$="" THEN GOTO 125
 130 CLS
 135 PRINT AT 1,9;"          ";AT
2,8;"         "
 140 PRINT AT 3,7;"          "
;AT 4,7;"          "
 145 PRINT AT 5,7;" *LEMONADE* "
;AT 6,7;"     OO     +"
 150 PRINT AT 1,1;"TIME"
 160 PRINT AT 2,2;":00"
 170 LET T=9
 180 PRINT AT 2,1;T;":00 "
 190 LET T=T+1
 200 IF T=13 THEN LET T=1
 210 IF T<>6 THEN GOTO 180
 220 PRINT AT 2,1;"CLOSED"
 230 FOR I=1 TO 100
 235 NEXT I
 240 CLS
 250 LET N=RND*CODE (A$(D,17))+R
ND*CODE (A$(D,18))
 260 LET A=INT (N-(1/(10-(L-P)))
*N)
 270 IF A>C THEN LET A=C
 280 PRINT "  SALES REPORT
          "
 290 PRINT
 300 PRINT "CUPS SOLD  ";A
 310 PRINT "MONEY IN THE TILL £
";(A*L)/100
 320 PRINT "WHOLESALERS BILL £"
;(C*P)/100
 325 PRINT "RENT £";R/100
 330 PRINT
 340 PRINT "PROFIT £";(L*A-C*P-
R)/100
 350 LET M=M+(A*L-C*P-R)
 360 PRINT AT 10,0;"PRESS A KEY
TO CONTINUE"
 370 IF INKEY$="" THEN GOTO 370
 375 CLS
 380 IF RND>.7 THEN LET P=P+1
 390 IF RND>.7 THEN LET R=R+5
 400 GOTO 30
```

```
         LEMONADE STAND
REPORT      HEATWAVE
LEMONADE    3 P PER CUP
MONEY       £10.41
130  CUPS MADE FOR TODAY

   SALE PRICE ?
```

TIME
5:00




```
  SALES REPORT

CUPS SOLD  109
MONEY IN THE TILL £5.45
WHOLESALERS BILL £3.9
RENT £0.55

PROFIT £1

PRESS A KEY TO CONTINUE
```

Spiralling into graphics

Keith Hewson, of Hornsea, has developed a fascinating Spirograph program for the 16K ZX81 — which even allows 'impossible' patterns to be generated.

Spirograph patterns are formed by both the interior and exterior epicycloid curves.
There are two basic equations for these curves:

INTERIOR EPICYCLOID:
X=(A-B) COS I+H COS ((A-B) I/B)
Y=(A-B) SIN I−H SIN ((A-B) I/B)
EXTERIOR EPICYCLOID:
X=(A+B) COS I−H COS ((A+B) I/B)
Y=(A+B) SIN I−H SIN ((A+B) I/B)
where A is the radius of the large circle (circle A), B is the radius of the small circle (circle B) and H is a point on the circumference of the small circle.

The epicycloid

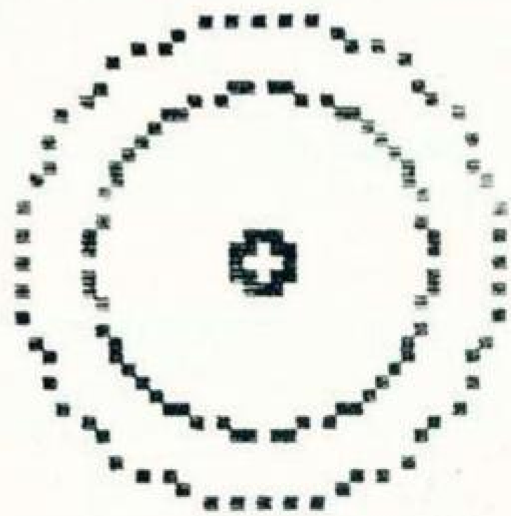
This is the curve traced out by the point H on the circumference of the small circle, which rolls on the inside or outside of the large circle. As with the commercial toy, 'Spirograph', the point H can be moved from the circumference to any point along the radius to the centre, where it will just draw a circle.

When the point H is nearer the circumference of the small circle, the patterns tend to be geometric in shape. When point H is moved nearer to the centre of circle B, the curve tends to be more floral, or organic in shape. The general program for the interior epicycloid is:

```
1 INPUT A
2 INPUT B
3 INPUT H
10 FOR I = 0 TO 2 *PI STEP PI/10
20 LET X=(A-B)*COS I+H *COS ((A-B)* I/B)
30 LET Y=(A-B)*SIN I-H*SIN ((A-B)*I/B
40 PLOT 30 + X, 22 + Y
50 NEXT I
```

Drawing patterns

The basic shapes of the hexagon, pentagon, square, triangle and ellipse can easily be entered into this program. Anything higher than a hexagon tends to become indistinct. To draw other patterns, the value for I has to be changed. It varies, on the ZX81, from two to 26 times PI.



There are a few ways of increasing the variety of patterns, but with low resolution graphics, the number is restricted. The three best ways of increasing the variety of patterns is:

1 'Moving to the centre', that is, moving the point H from the circumference of circle B to near its centre. Try this, with A set equal to 20 and B to 10.

```
3 FOR H=2 TO 10 STEP 4
60 NEXT H
```

2 'Moving to the side', that is, rotation of the pattern, which can be done by moving an angle, which must be added to the general program. Try it, with A equal 20, B equal 10, H equal to 8, and with the I loop FOR I 0 TO 2*PI STEP PI/10. You'll need to add the following:

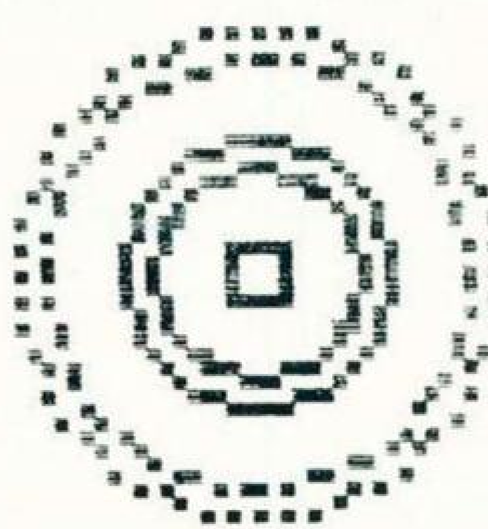
```
4 FOR J=0 TO 3*PI/2
20 LET X=(A-B)*COS I+H*COS((A-B)*I/B+J)
30 LET Y=(A-B)*SIN I+H*SIN((A-B)*I/B+J)
60 NEXT J
```

3 'Moving to the centre and the side', that is, movement of point H and rotation. The program is as in two, with the addition of:

```
3 LET H=0
6 LET H=H + 2
```

There is also the 'extended internal pattern' which is achieved by adding the following to the general program:

```
3 LET F=10
8 FOR H=2 TO 8 STEP 2
9 LET F=F + 6
40 PLOT F + X, 22 + Y
60 NEXT H
```



```
 1 INPUT A
 2 INPUT B
 3 INPUT H
 4 FOR J=0 TO 3*PI/2
10 FOR I=0 TO 2*PI STEP PI/30
20 LET X=(A-B)*COS I+H*COS ((A
-B)*I/B+J)
30 LET Y=(A-B)*SIN I+H*SIN ((A
-B)*I/B+J)
40 PLOT 30+X,22+Y
50 NEXT I
60 NEXT J
```

Ratios

Although the low resolution graphics tend to be limiting, the computer offers experimentation in other directions, which are not available on a real Spirograph. Besides the obvious one of a greater variety of ratios, there is the possibility of putting the smaller number in first, and the bigger number in second, as though the larger circle is rolling inside the smaller one.

The following ratios will fit easily into the general program:

A	4	5	6	7	8	9
B	13	13	13	13	13	13
H	10	10	10	10	10	10

And for these use FOR I=0 TO 26*PI

A	4	7	8
B	9	9	9
C	8	8	8

For these use FOR I=0 TO 18*PI

A	4	6	8
B	11	11	11
H	9	9	9

And for these we suggest FOR I=0 TO 22*PI

Some of these have an almost mandala-like quality.

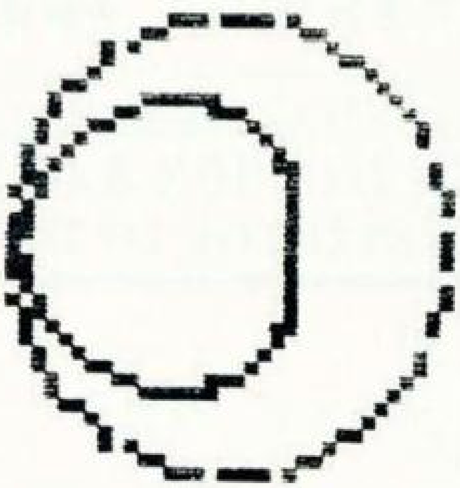
The following can be used in the exterior equations:

A	4	5	6
B	7	7	7
H	6	6	6

And use FOR I=0 TO 14*PI

A 4 5
B 9 9
H 6 7

FOR I=0 TO 18*PI





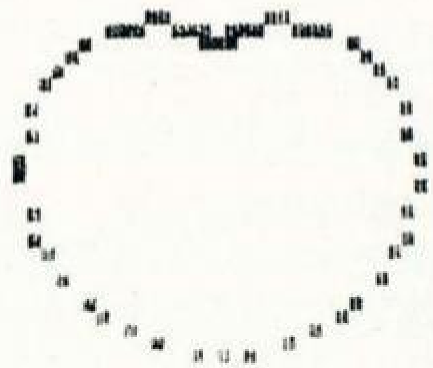
Experiments

Another direction for experimentation is based on the moving point H, program one. Put the H loop into the I loop. This will draw the points of H first (FOR H=0 TO 6).

And finally, a way of showing the smaller circle moving inside the larger circle, by using the rotation program, is as follows. Put the J loop inside the I loop. This will draw just six circles. Use FOR I=0 TO 2*PI STEP PI/10 and FOR J=0 TO 2*PI STEP PI/10.







MEMOTECH Explores the Excellence of your ZX81

Access Barclaycard Holders
Please Tel. Oxford
(0865) 722102



NEW - Memotech's Plug-in ZX81 Keyboard

- Plugs directly into the back of your ZX81
- Keys have durable Sinclair legends
- High quality typewriter keys
- Does not inhibit other add-ons
- Automatic hold-down repeat
- Complete with buffered interface
- Fast and easy data entry
- Moveable between configurations

Memotech's Memopak Range

Current Memopaks
16K
32K
64K
HRG
Centronics Interface

MEMOPAK 64K MEMORY EXTENSION

The 64K Memopak extends the memory of the ZX81 by 56K, and with the ZX81 gives 64K, which is neither switched nor paged and is directly addressable. The unit is user transparent and accepts commands such as 10 DIM A(9000).
Breakdown of memory areas...0-8K Sinclair ROM. 8-16K This area can be used to hold machine code for communication between programmes or peripherals. 16-64K A straight 48K for normal Basic use.

MEMOPAK 32K and 16K MEMORY EXTENSIONS

These two packs extend and complete the Memotech RAM range (for the time being!) A notable feature of the 32K pack is that it will run in tandem with the Sinclair 16K memory extension to give 48K RAM total.

MEMOPAK HIGH RES GRAPHICS PACK

HRG Main Features — • Fully programmable Hi-Res (192 × 248 pixels) • Video page is both memory and bit mapped and can be located anywhere in RAM. • Number of Video pages is limited only by RAM size (each takes about 6.2K RAM) • Instant inverse video on/off gives flashing characters • Video pages can be superimposed • Video page access is similar to Basic plot/unplot commands • Contains 2K EPROM monitor with full range of graphics subroutines controlled by machine code or USR function

MEMOPAK CENTRONICS TYPE PARALLEL PRINTER INTERFACE

Main Features — • Interfaces ZX8I and parallel printers of the Centronics type • Enables use of a range of dot matrix and daisy wheel printers with ZX8I • Compatible with ZX8I Basic, prints from LLIST, LPRINT and COPY • Contains firmware to convert ZX8I characters to ASCII code • Gives lower-case characters from ZX8I inverse character set



© R. Branton & G.A.C. Boyd 1982

Coming Soon..

RS232 Interface

Digitising Tablet

We regret we are as yet unable to accept orders or enquiries concerning the above products, but we'll let you know as soon as they become available.

Please make cheques payable to **Memotech** Ltd.

Please send me	Price	No	Total
64K RAM £68.70+£10.30 VAT	£79.00		
32K RAM £43.43+£6.52 VAT	£49.95		
16K RAM £26.00+£3.90 VAT	£29.90		
HRG £52.00+£7.80 VAT	£59.80		
CENTRONICS I/F £34.70+£5.20 VAT	£39.90		
MEMOTECH KEYBOARD. Price £43.43+£6.52 VAT	£49.95		
Packaging & Postage £2.00 per unit (UK), £3.00 (overseas)			

TOTAL ENC

Please Debit my Access/Barclaycard* account number

*Please delete whichever does not apply

SIGNATURE ________ NAME ________

ADDRESS ________ DATE ________

Now Available at major branches of WHSMITH

We want to be sure you are satisfied with your Memopak — so we offer a 14-day money back Guarantee on all our products.
Memotech Limited, 3 Collins Street, Oxford OX4 1XL, England Tel: Oxford (0865) 722102 Telex: 837220 Orchid G

Wage control for a small firm



From Abingdon, Alan Beadle sent us this program which has been used in a small hardware shop owned by Alan and his two brothers

Payslips

The program calculates and prints Payslips and a Wages Record (used as a Wages Book). It calculates the date and at the end of each Financial Month it shows the amounts which have to be paid to Inland Revenue and National Insurance. It also provides running annual totals.

The Payslips are deliberately comprehensive and the annual totals on them are copied by hand on to the Form P11, (the Deductions Working Sheet), which has to be filled in and kept. Columns 4 and 5 on this form are not available, but they are only there for calculating the essential figures and I leave them blank.

A list of variables is given in Fig. 1. Where possible I try to give one variable the initials of the function it represents: e.g. NEA is National Insurance rate for employee Table A.

The program is saved and run by lines 9000 to 9020. If successful then Fig. 2 will appear. At the start of a new year the program will have to be STOPped at this point, and the data in lines 1300 to 1400 checked. D in line 1320 is the date of the first payday in the year. RUN is now used for the only time. Fig. 2 again shows and E is used to input 1st week's data as requested. Fig. 3 should be a result. When all employees' data is in, NEWLINE will print the first payslip. If all is satisfactory, it then can be transferred to the printer by inputting Z (the COPY key). NEWLINE or Z are now used until the week's run is completed. A further NEWLINE gives the choice of continuing for another week or saving.

COPY rather than LPRINT has to be used because of the method used to align columns and print zeros after the decimal point. The way it works can be seen if the program is run in SLOW.

Dates are complicated, in that the Financial Year begins on April 6th, and each ensuing month on the 6th of that month. The date calculations are in lines 4500 to 4680. P is a marker used to decide whether the month's totals are to be printed. If they are to be then lines 4690 to 4740 set up the next month. They extract the number of days in the month from M$ and also check for a leap year.

Calculating taxes

New Tax Tables came into force for the 4th. week of 1982-1983 tax year. The first three weeks are calculated in lines 5050 to 5090. These lines can be removed after week three, but *MUST* be removed for subsequent years. Remove the lines by INPUTting S when Fig. 2 is on the screen, deleting the lines, and GOing to 2000, when Fig. 2 will re- appear.

Only 32 columns on the printer is somewhat restrictive and it is this, in fact, which limits the number of employees to 9. There is not sufficient room to print names and all the data necessary for the Wages Book on one line. The names are therefore listed with their numbers (Fig. 4), and this used as a reference in Fig. 5. Even an employee number of 10 in Fig. 5 would require 33 columns. Using letters instead of numbers as cross-reference would allow for up to 26 employees.

No entry of overtime is put in the programme. We do not get involved with it and the program as here already takes three and a half minutes to load or save. The program line numbers have been deliberately spread to allow for such extras. 3300 to 3500 and 4000 to 4500 are available.

The only instruction not shown on screen is the use of Z to copy to the printer, but one does not want this to appear on each Payslip.

Finally, at the end of the Financial Year, an extra NEWLINE will print the total National Insurance paid (including the Employer's portion) for each employee. This is needed for end-of-year Tax Returns.

Variables

M$	Days in months
N$	Firm's name
R$	Firm's Tax reference number
W$	"WAGES"
P$	".00"
A$()	Employee's name
B$()	Employee's Tax Code (including letter)
C$()	Employee's Insurance number
A()	Employee's pay per week
B()	Employee's Tax number (B$() less its letter)
C()	Employee plus employer Nat. Ins. per week
D()	Employee's Nat. Ins. per week
E()	Employee's total pay for year
F()	Employee plus employer Nat Ins total for year
G()	Employee's Nat. Ins. total for year
X()	Employee's total tax due to date
Y()	Employee's total tax paid up to last week
Z()	Employee's tax due this week (Z() minus Y())

TR	Tax rate
TT	Tax threshold
NEA	Employee's Nat. Ins. rate (Table A)
NEB	Employee's Nat. Ins. rate (Table B)
NR	Employee's Nat. Ins. rate
YY	Year
MA	Calendar month
MF	Financial month
W	Week
D	Payday
P	Next Week's Payday
WP/WT/WN	Weekly totals of Nett Pay/Tax due/Nat. Ins. due
MP/MT/MN	Monthly totals of Nett Pay/Tax/Nat. Ins. due
YP/YT/YN	Yearly totals of Nett Pay/Tax/Nat. Ins. due
M	Number of days in current month.

Other letters used as temporaries are B E F G L S T V

Table 1: Important line numbers

1300	Year
1310	Days in Months
1320	First Pay Day
1360	Tax Rate
1370	Employee Ins. Rate (standard)
1375	Employee Ins. Rate (reduced)
1380	Employer Ins. Rate
1390	Tax Thresh-hold
2000	prepare for next week
2500	change employee's pay or code
3000	change Tax or Insurance Rates
3500	data for new employee
4500	increment by 1 week
5000	calculate figures for week
5350	print Payslip
7000	print names and tax codes and employee number
7080	print totals for Wage Book
8520	print extra figures needed at end of year
8890	SAVE routine



```
  20 REM  A$()=NAME B$()=TAX CODE
  30 REM  C$()=NAT  INS.  NUMBER
  40 REM  A()=PAY/WEEK
          C()=WEEK  INS.(BOTH)
          D()=WEEK  INS.(EMPLOYEE)
          E(J)=TOTAL  WAGE(YEAR)
          F()=TOTAL  INS.(BOTH)
          G()=TOTAL  INS.(EMPLOYE)
  50 REM
        TR=TAX  RATE
        NE=NAT  INS  RATE(EMPLOYEE)
        NR=NAT  INS  RATE(EMPLOYER)
        TT=TAX  THRESHHOLD
  60 REM
 100 LET  N$="BEADLES"
 110 LET  W$="WAGES"
 120 LET  R$="OX3 547/B62"
 130 LET  P$=".00"
 200 DIM  A$(9,12)
 210 DIM  B$(9,5)
 220 DIM  C$(9,13)
 300 DIM  A(9)
 310 DIM  B(9)
 320 DIM  C(9)
 330 DIM  D(9)
 340 DIM  E(9)
 350 DIM  F(9)
 360 DIM  G(9)
 370 DIM  X(9)
 380 DIM  Y(9)
 390 DIM  Z(9)
1300 LET  YY=1982
1310 LET  M$="303130313130313031312831"
1320 LET  D=9
1330 LET  W=0
1340 LET  MF=1
1350 LET  MA=4
1360 LET  TR=.3
1370 LET  NEA=8.75
1380 LET  NEB=3.2
1390 LET  NR=13.7
1400 LET  TT=30
1410 LET  YP=0
1420 LET  YT=0
1430 LET  YN=0
1440 LET  J=0
2000 CLS
2010 PRINT  TAB 8;N$
2020 PRINT
2030 PRINT  TAB 8;W$
2040 PRINT
2050 PRINT  TAB 4;"NEXT WEEK ";W+1
2060 PRINT
2070 PRINT  """""" TO RUN"
2080 PRINT  "T FOR TAX OR N.I. CHANGE "
2085 PRINT  "V FOR EMPLOYEE DATA CHANGE"
2090 IF W>0 THEN PRINT "E FOR NEW EMPLOYEE "
2095 IF W=0 THEN PRINT "E FOR DATA FOR NEXT YEAR"
2100 PRINT  "S TO STOP"
2110 PRINT  ,,,,"REM Z WHEN DATA SHOWING TO          PRINTOUT"
2140 INPUT  Q$
2150 IF  Q$="S"  THEN  STOP
2160 IF  Q$="V"  THEN  GOTO  2500
2170 IF  Q$="E"  THEN  GOTO  3500
2175 IF  Q$="T"  THEN  GOTO  3000
2180 IF  Q$=""  THEN  GOTO  4000
2190 GOTO  2000
2500 CLS
2510 PRINT  "EMPLOYEE NUMBER? ";
2520 INPUT  J
2530 PRINT  J
2540 PRINT
2550 PRINT  "WAGE ";A(J);"        CODE   ";B$(J)
2560 PRINT
2570 PRINT  "NEW WAGE? ";
2580 INPUT  A(J)
2600 PRINT  A(J)
2910 PRINT
2920 PRINT  "NEW CODE? ";
2930 INPUT  B$(J)
2950 PRINT  B$(J)
2960 PRINT
2970 PRINT  "NEWLINE TO CONTINUE"
2980 INPUT  Q$
2990 GOTO  2000
3000 CLS
3110 PRINT  "TAX RATE ";TR
3120 PRINT  "NEW RATE? ";
3130 INPUT  TR
3140 PRINT  TR
3150 PRINT
3160 PRINT  "INS RATE (EMPLOYEE) ";NE
3170 PRINT  "NEW RATE? ";
3180 INPUT  NE
3190 PRINT  NE
3200 PRINT
3210 PRINT  "INS RATE (EMPLOYER) ";NR
3220 PRINT  "NEW RATE? ";
3230 INPUT  NR
3240 PRINT  NR
3250 GOTO  2960
3500 CLS
3505 IF W=0 THEN PRINT "NEXT AVAILABLE EMPLOYEE NO.";J+1,,,
3510 PRINT  "EMPLOYEE NUMBER? ";
3520 INPUT  J
3530 PRINT  J
3540 IF  CODE A$(J)=0 THEN GOTO 3630
3550 PRINT
3560 PRINT  "ALREADY USED  ";A$(J)
3570 INPUT  Q$
3580 GOTO  3500
3630 PRINT
3640 IF W>0 THEN PRINT "DATA FROM FORM P35 WEEK ";W
3650 PRINT
3670 PRINT  "NAME? ";
3680 INPUT  A$(J)
3690 PRINT  A$(J)
3700 PRINT
3710 PRINT  "TAX CODE? ";
3720 INPUT  B$(J)
3730 PRINT  B$(J)
3740 PRINT
3750 PRINT  "NAT.INS.NO.? ";
3760 INPUT  C$(J)
3770 PRINT  C$(J)
```

```
3775 IF W=0 THEN GOTO 3860
3780 PRINT
3790 PRINT "TOTAL WAGES TO DATE?
";
3800 INPUT E(J)
3810 PRINT E(J)
3820 PRINT
3830 PRINT "TAX TO DATE? ";
3840 INPUT Y(J)
3850 PRINT Y(J)
3860 PRINT
3870 PRINT "WAGE= ";
3890 INPUT A(J)
3900 PRINT A(J)
3910 GOTO 2960
4500 LET D=D+7
4505 IF W=0 THEN LET D=D-7
4507 LET W=W+1
4510 LET P=D+7
4520 LET WP=0
4530 LET WT=0
4540 LET WN=0
4550 IF W=1 THEN GOTO 4640
4570 IF P>M THEN LET P=P-M
4580 IF D<=M THEN GOTO 4620
4590 LET MA=MA+1
4600 IF MA>12 THEN LET MA=1
4610 LET D=D-M
4620 IF D<5 OR D>11 THEN GOTO 50
00
4630 LET MF=MF+1
4640 LET M=VAL M$(MF*2-1 TO MF*2
)
4650 LET MP=0
4660 LET MT=0
4670 LET MN=0
4680 IF MF=10 THEN LET YY=YY+1
4690 IF MF=11 AND YY=(INT (YY/4)
)*4 THEN LET M=29
5000 FOR J=1 TO 9
5010 LET E(J)=E(J)+A(J)
5020 LET B=0
5030 IF A(J)<TT AND Y(J)=0 THEN
GOTO 5180
5040 IF B$(J, TO 2)<>"W1" THEN L
ET BT=VAL (B$(J, TO 3))
5050 IF W>3 THEN GOTO 5100
5060 IF B$(J, TO 2)="W1" THEN LE
T BT=137
5070 LET B=INT ((((BT*10)+11.5)/
5.2)+.5)/10
5080 IF B*52>(BT*10)+11.5 THEN L
ET B=B-.05
5090 GOTO 5120
5100 IF B$(J, TO 2)="W1" THEN LE
T BT=156
5110 LET B=INT (((BT+.899)*1000)
/52)/100+.01
5120 IF INT A(J)>INT (A(J)-.5) T
HEN LET A=INT A(J)+.25
5130 IF INT A(J)=INT (A(J)-.5) T
HEN LET A=INT A(J)+.75
5140 LET D(J)=INT (A*NEA+.5)/100
5145 IF B$(J,5)="B" THEN LET D(J
)=INT (A*NEB+.5)/100
5150 LET C(J)=INT (A*NR+.5)/100+
D(J)
5152 IF A(J)<TT THEN LET C(J)=0
5154 IF A(J)<TT THEN LET D(J)=0
5160 LET G(J)=G(J)+D(J)
5170 LET F(J)=F(J)+C(J)
5190 IF B$(J, TO 2)="W1" OR W=53
 THEN GOTO 5230
5200 IF A(J)>TT THEN LET X(J)=(I
NT ((E(J)-B*W)))*TR
5205 IF X(J)<0 THEN LET X(J)=0
5210 LET Z(J)=X(J)-Y(J)
5220 GOTO 5270
5230 LET Z(J)=0
5232 LET X(J)=Y(J)
5235 IF A(J)>TT THEN LET Z(J)=IN
T (((INT (A(J)-B))*TR)*100+.5)/1
00
5240 LET X(J)=X(J)+Z(J)
5270 LET Y(J)=X(J)
5280 LET F=A(J)-D(J)-Z(J)
5290 LET WP=WP+F
5300 LET WT=WT+Z(J)
5310 IF A(J)>0 THEN LET WN=WN+C(
J)
5320 LET Y(J)=X(J)
5330 CLS
5340 IF A(J)=0 AND W<52 THEN GOT
O 5800
5350 PRINT "WEEK ";W;"     PAYSLI
P    ";D;".";MA;".";YY
5360 PRINT
5370 PRINT TAB 9;"█";A$(J);"█"
5380 PRINT AT 5,3;"DEDUCTIONS";T
AB 21;"WAGES"
5390 PRINT AT 6,23;P$;AT 6,23-LE
N STR$ INT A(J);A(J)
5500 PRINT "TAX";AT 7,6;P$;AT 7,
6-LEN STR$ INT ABS Z(J);ABS Z(J)
5505 IF LEN STR$ Z(J)>LEN STR$ A
BS Z(J) THEN PRINT AT 7,10;"R"
5510 PRINT "INS";AT 8,6;P$;AT 8,
6-LEN STR$ INT D(J);D(J);AT 8,15
;"LESS"
5520 LET G=Z(J)+D(J)
5525 IF LEN STR$ G>LEN STR$ ABS
G THEN PRINT AT 8,27;"R";AT 10,1
0;"R"
5530 PRINT AT 8,23;P$;AT 8,23-LE
N STR$ INT ABS G;ABS G
5540 PRINT TAB 4;"-----
-----"
5550 PRINT AT 10,6;P$;AT 10,6-LE
N STR$ INT ABS G;ABS G;AT 10,15;
"NETT"
5560 LET G=A(J)-G
5570 PRINT AT 10,23;P$;AT 10,23-
LEN STR$ INT G;G
5580 PRINT TAB 4;"-----
-----"
5690 PRINT AT 13,5;"ANNUAL TOTAL
S"
5700 PRINT "PAY",P$;AT 14,16-LEN
 STR$ INT E(J);E(J)
5710 PRINT "TAX",P$;AT 15,16-LEN
 STR$ INT X(J);X(J)
5720 PRINT "INSURANCE",P$;AT 16,
16-LEN STR$ INT G(J);G(J)
5730 PRINT AT 18,2;"TAX CODE ";B
$(J, TO 4)
5740 PRINT "INS.NUMBER ";C$(J)
5750 PRINT
5760 PRINT TAB 4;N$;TAB 12;R$
5770 INPUT Q$
5780 IF Q$="Z" THEN COPY
5790 IF Q$="S" THEN STOP
5800 NEXT J
7000 CLS
7010 PRINT TAB 6;"NAME        CO
DE"
7020 FOR J=1 TO 9
7030 IF A(J)<>0 OR W>51 THEN PRI
NT CHR$ (J+156);"= ";A$(J);"
;B$(J)
7040 NEXT J
7050 INPUT Q$
7060 IF Q$="Z" THEN COPY
7070 CLS
7080 PRINT TAB 2;"WEEK ";W,D;"."
;MA;".";YY
7090 PRINT AT 3,2;"WAGES  TAX
INSURANCE  NETT"
7100 PRINT "               (EMP █
TOTAL)"
7110 LET L=4
7120 FOR J=1 TO 9
7130 IF A(J)=0 THEN GOTO 7160
7140 LET L=L+1
7150 PRINT AT L,0;CHR$ (J+156);A
T L,4;P$;AT L,4-LEN STR$ INT A(J
);A(J);AT L,10;P$;AT L,10-LEN ST
```

```
R$ INT Z(J);Z(J);AT L,16;P$;AT L
,16-LEN STR$ INT D(J);D(J);AT L,
22;P$;AT L,22-LEN STR$ INT C(J);
C(J);AT L,29;P$;AT L,29-LEN STR$
 INT (A(J)-Z(J)-D(J));A(J)-Z(J)-
D(J)
7160 NEXT J
7600 LET L=L+2
7610 PRINT AT L,12;"TOTALS"
7620 LET L=L+1
7630 PRINT AT L,7;"TAX       INSUR
ANCE      NETT"
7640 LET L=L+1
7650 PRINT AT L,0;"WEEK";AT L,9;
P$;AT L,9-LEN STR$ INT WT;WT;AT
L,19;P$;AT L,19-LEN STR$ INT WN;
WN;AT L,29;P$;AT L,29-LEN STR$ I
NT WP;WP
7660 LET MP=MP+WP
7670 LET MT=MT+WT
7680 LET MN=MN+WN
7690 LET YP=YP+WP
7700 LET YT=YT+WT
7710 LET YN=YN+WN
7720 IF P<5 OR P>11 THEN GOTO 88
70
7730 LET L=L+1
7740 PRINT AT L,0;"MONTH";AT L,9
;P$;AT L,9-LEN STR$ INT MT;MT;AT
 L,19;P$;AT L,19-LEN STR$ INT MN
;MN;AT L,29;P$;AT L,29-LEN STR$
INT MP;MP
7750 LET L=L+1
7760 PRINT AT L,0;"YEAR";AT L,9;
P$;AT L,9-LEN STR$ INT YT;YT;AT
L,19;P$;AT L,19-LEN STR$ INT YN;
YN;AT L,29;P$;AT L,29-LEN STR$ I
NT YP;YP
7770 IF MF<12 THEN GOTO 8870
8510 INPUT Q$
8520 CLS
8530 PRINT "NAME           NAT.INS.
TOTAL"
8540 LET L=2
8550 FOR J=1 TO 9
8560 LET L=L+1
8570 IF F(J)>0 THEN PRINT AT L,0
;A$(J);AT L,18;P$;AT L,18-LEN ST
R$ INT F(J);F(J)
8580 NEXT J
8860 STOP
8870 INPUT Q$
8880 IF Q$="Z" THEN COPY
8890 CLS
8900 PRINT """C"" FOR ANOTHER WE
EK"
8910 PRINT """R"" TO SAVE"
8920 PRINT """S"" TO STOP"
8930 INPUT Q$
8940 IF Q$="C" THEN GOTO 2000
8950 IF Q$="R" THEN GOTO 8980
8960 IF Q$="S" THEN STOP
8970 GOTO 8890
8980 PRINT "START RECORDER AND P
RESS ""R"" AND NEWLINE"
8990 INPUT Q$
9000 SAVE "WAGES"
9010 CLS
9020 GOTO 2000
```

Fig. 1 — sample payslip

```
WEEK 4      PAYSLIP      30.4.1982

            C.C.COX

   DEDUCTIONS           WAGES
                       106.68
TAX  17.40
INS   9.34       LESS   26.74
     -----              -----
     26.74       NETT   79.94
     -----              -----

        ANNUAL TOTALS
PAY             426.72
TAX              68.70
INSURANCE        37.36

   TAX CODE 256H
INS.NUMBER CC/77/88/99/C

     BEADLES OX3 547/862
```

Fig. 2

```
           BEADLES

            WAGES

      NEXT WEEK 5

"" TO RUN
T FOR TAX OR N.I. CHANGE
U FOR EMPLOYEE DATA CHANGE
E FOR NEW EMPLOYEE
S TO STOP

REM Z WHEN DATA SHOWING TO
    PRINTOUT
```

Fig. 3

```
NEXT AVAILABLE EMPLOYEE NO.1

EMPLOYEE NUMBER? 1

NAME? A.A.ALAN

TAX CODE? 137L

NAT.INS.NO.? AA/11/22/33/A

WAGE= 65.25

NEWLINE TO CONTINUE
```

Fig. 4

```
         NAME               CODE
 =  A.A.ALAN                137L
 =  B.B.BROWN               213H
 =  C.C.COX                 256H
 =  D.D.DAVIS               156L
 =  E.E.EAST                137L
 =  G.G.GREEN               186H
```

Fig. 5

```
  WEEK 4               30.4.1982

 WAGES   TAX     INSURANCE      NETT
                (EMP   TOTAL)
  65.25  11.40   5.71  14.65   48.14
  76.30  10.50   6.67  17.12   59.13
 106.68  17.40   9.34  23.96   79.94
 123.45  28.20  10.78  27.67   84.47
  20.00   0.00   0.00   0.00   20.00
  88.88  15.90   7.77  19.93   65.21

                TOTALS
         TAX     INSURANCE      NETT
WEEK     83.40    103.33      356.89
MONTH   331.80    413.32     1429.36
YEAR    331.80    413.32     1429.36
```

KRAK!

THE SPECTRUM 'ADD-ON'

ONLY £19.50 + VAT

BLADAMM!!

COMPLETE your SPECTRUM with our Multi-purpose Sound Generator and Joystick-port Board!! With one low-cost purchase you can obtain the following outstanding improvements to your ZX SPECTRUM:

- THREE CHANNEL sound effects. PROGRAM three independent sounds with music, gun shots, explosions etc.
- AMPLIFICATION of the standard sound output.
- PROVISION of JOYSTICK PORTS to allow simple connection of one or two suitable joysticks (see below).

The 'ADD-ON' uses the amazing AY-3-8910 SOUND Chip, which gives you an enormous range of sound effects. The output and volume of each channel can be separately controlled, with gun shots, explosions, drums etc., produced under the 'sound envelope'. MOST IMPORTANT TO NOTE—the CHIP is 'processor independent.' This means that sounds can continue to be produced without any noticeable effect on the speed of your program!!

The 'ADD-ON' contains its own speaker and amplifier chip and is supplied with a cassette containing sound demonstration routines and some sample sounds.

The 'ADD-ON' simply plugs straight onto the back of your SPECTRUM. No soldering is required.

The 'ADD-ON' is offered at the incredibly low price of £19.50 + VAT.

COMPLETE YOUR SPECTRUM NOW!!

JOYSTICK KITS £3.95 + VAT
To complement our exciting new 'ADD-ON' for the SPECTRUM we are offering low-cost Joystick Kits. Each one contains a potentiometer-joystick, a fire-button and a suitable connector.

NEW ORIGINAL SPECTRUM SOFTWARE
—from PROGRAM POWER—using the 'ADD-ON' to great effect!!

CABMAN (Machine Code) £3.95 + VAT
You are the owner of a fleet of yellow taxis. A rival firm competes with you, trying to steal your fares and run you off the road, if they can. Ten skill levels determine the number of opposing taxis, the speed at which they drive and the amount you earn per completed journey. MAXIMISE your earnings before you lose your entire fleet. This program is FIRST CLASS.

ATTENTION MACHINE CODE PROGRAMMERS!
We can now offer SPECTRUM versions of the ASSEMBLER and DISASSEMBLER programs from ACS Software. The ZX81 versions have proved extremely popular.

ULTRAVIOLET (Assembler) ________ £6.95 + VAT
INFRARED (Disassembler) ________ £5.95 + VAT

BOOKS

SPECTRUM
Over the Spectrum ________ £6.95
Cambridge Colour Collection ________ £6.95

ZX81
Mastering Machine Code on your ZX81 __ £7.50
NOT ONLY 30 Programs ________ £6.95
Getting Acquainted with your ZX81 ____ £5.95
49 Explosive Games for the ZX81 ______ £5.95
Making the most of your ZX81 ________ £5.95
Byteing Deeper into your ZX81 ________ £4.95
The Explorer's Guide to the ZX81 ______ £4.95

BBC, DRAGON & ACORN DEALERS

NOTE: We stock (and market) PROGRAM POWER SOFTWARE for the BBC, ACORN, SPECTRUM and DRAGON 32. Send S.A.E. for appropriate catalogue.

MICRO POWER
8/8a REGENT STREET,
CHAPEL ALLERTON,
LEEDS LS7 4PE.
Tel: (0532) 683186

All programs are now available at all good dealers or direct from MICRO POWER LTD.

Please add 55p order P & P + VAT at 15%.

VISA

WRITTEN ANY PROGRAMS? WE PAY 20% ROYALTIES FOR DRAGON, SPECTRUM BBC, ATOM PROGRAMS

PROGRAM POWER MICRO

V & H COMPUTER SERVICES PRESENTS

SPECTRUM SPECTACULAR

(50 Programs for the Sinclair Spectrum)

And what programs! New games! Old games! Business programs (like invoicing and sales ledger), word processor, machine code routines, graphics....

All in **SPECTRUM SPECTACULAR** the new book by Roger Valentine.
SPECTACULAR VALUE AT £4.95

By the same author:—

WHAT CAN I DO WITH 1K (ZX81)........... £4.95
WHAT CAN I DO WITH 16K (ZX81)......... £4.95

V & H COMPUTER SERVICES
182c KINGSTON ROAD,
STAINES, MIDDX.
TEL: STAINES 58041

Business users please note: Our PAYROLL program is now available for ZX81, SPECTRUM, PET, BBC and KONTRON. Please write for details or send £2.00 for comprehensive manual.

GAMES CENTRE

THE LARGEST SELECTION OF GAMES IN THE WORLD

We stock the **BIG NAMES** in Computers including

ATARI 400/800
SINCLAIR ZX 81
ZX SPECTRUM
VIC-64
DRAGON MICRO

and a wide range of independent SOFTWARE

Main Computer Branches: 22 OXFORD STREET, London W.1.
439 OXFORD STREET, London W.1.
52 WESTERN ROAD, BRIGHTON.

Also at: 184 REGENT STREET, London W.1.
254 REGENT STREET, London W.1.

What's it all about, Michael?

Mike Salem, head of the ZX business software firm Hilderbay, takes on the difficult task of explaining what a computer really is, and how it works. He also takes a look at the use of discs and cassettes for external memory.

Techniques

I believe that you don't, and shouldn't, need to learn about programming to make effective use of a computer, any more than you need to be able to manufacture and repair your car to make good use of it. Of course, there are advantages in being able to do everything yourself. After you have learnt the necessary techniques, you'll be able to make your computer or car do exactly what you want it to do, and you'll not depend upon people of various degrees of competence and availability to get things done. Having said this, you will find that even software professionals prefer to buy, rather than write, software.

Processing unit

In its most primitive form, a computer is a device which operates upon numbers under the control of other numbers. It possesses a processing unit (CUP, central processing unit) which actually carries out the numerical operations, a memory where a mixture of instructions and data is stored, and some way of getting information in from the outsideworld, and of sending results out.

ROM

Computers always contain some memory which can be copied (read) only, and not changed (written to). This sort of memory is known, fairly logically, as Read Only Memory (ROM).

The BASIC interpreter (to be described shortly), is stored in ROM in many machines, including the Spectrum and ZX81.

Computers must always have memory which can be both read and written to. It is possible to access any memory location of both types of memory, immediately, without scanning through all the memory in order. This is called random access. Read/write memory is called Random Access Memory (RAM). So ROM is fixed memory, and tells the computer how to do what it needs to do when running, and RAM is changeable memory which holds the program you are currently running and the intermediate results of that program.

Memory cell

If the sequence of numbers '42 0 80 237 75 2 80 237 74 34 0 80' was presented to the CPU of your Spectrum, the computer would copy the number in memory cell 20480 into the CPU, add to it the number in cell 20483, and store the result in cell 20480 (discarding the original contents). Other computers would interpret the results quite differently. In the BASIC computer language, which your ZX computer uses, this sequence of operations would be coded LET A = A + B.

Language

After a few years of translating human-language problems into sequences of numbers, some bright spark had the idea of writing a program, (as a sequence of numbers!), which would read a sequence of letters such as LET A = A + B and automatically translate it into '42 0 80 237 75 2 80 237 74 34 0 80". So, this translation program was written, and from then on everyone could write programs in the more comprehensible form, allowing the computer to translate it into its own language.

Translation programs were written for many different computers, so the line LET A = A + B would be changed into a different sequence of numbers, depending on the computer being used. The point is that the BASIC language itself is a computer program. When you program a computer in BASIC, you are really user of a program called an interpreter.

Why not go one level higher? A program designed to carry out a specific task can be considered to be a kind of language. The program itself can be written either in BASIC (or a similar, human-like language), or directly in number-sequence lanuage (known as machine language).

Basic interpreter

When the BASIC interpreter is command, the key will have a different meaning. For example, it may cause somebody's gross pay to be automatically calculated from the net pay (it is more usual to compute the net from gross). It is conceivable that a business program system could be written which would call a payroll program as part of itself. In this sort of hierarchal nest of structures, it is necessary for the user to be familiar only with the most general, the highest level, usage. Each level becomes more specific.

Machine language allows one to make the computer do anything that it conceivably can do. BASIC allows it to do most things, but is less flexible (albeit much easier to use). A chess program allows a computer only to play chess, until a different program is loaded. In some cases, a computer has a program more specific than BASIC permanently built in. This is the case with dedicated word processors and such things as washing machine controllers.

The common insistence that one should use a computer to learn to use the BASIC interpreter program, that is to learn programming in BASIC, misses the point of what a computer is. Using the same reasoning, why the BASIC level, rather than machine language? If you use a dedicated word processor, you don't worry about its internal workings. Why shouldn't you treat a computer running a word processor program in the same way?

In general, why not learn to use a system comprising a computer, and a set of peripherals and software, that make it do what you want it to do, rather than learning BASIC? Having said all this, we are the first to agree that there are many valid reasons for learning BASIC and similar languages, but you shouldn't feel that you must.

Storage

The general term 'mass storage devices' (MSD) is used to refer to means of holding programs outside the RAM

area of the computer. The cheapest MSD is the humble domestic portable monophonic cassette recorder. There are also 'floppy tapes', controllable cassette tape drives, and various other devices with size and price more suitable to large computers. Tape is cheap, slow and inherently sequential. That is, you have to go through it to find your place. The more specialised tape devices let the computer search through the tape to find the required item. You have to press the buttons on a domestic recorder.

Floppy discs

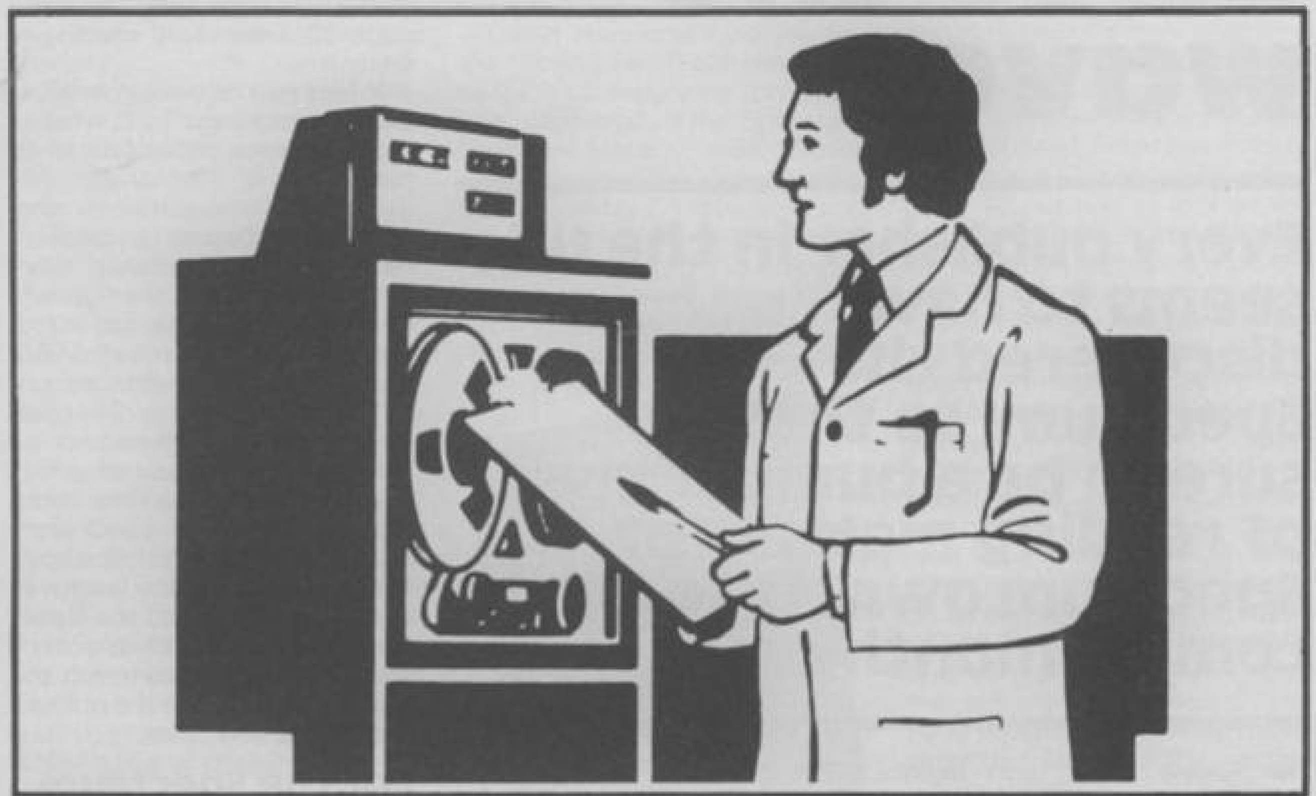
The other main group of MSDs is the disc family. These range from the Sinclair Microdrives, through conventional floppy discs, up to hard discs. Floppy discs are flexible discs coated with magnetic material. They are permanently sealed in square envelopes within which they are free to rotate. They are used in conjunction with 'floppy disc drives', which contain magnetic heads which can detect and create magnetic patterns on the discs. Hard discs can hold more information than can floppies, are much faster, and much more expensive. A typical floppy can hold between 100K and 800K. Hard discs can hold hundreds of thousands of kilobytes of information.

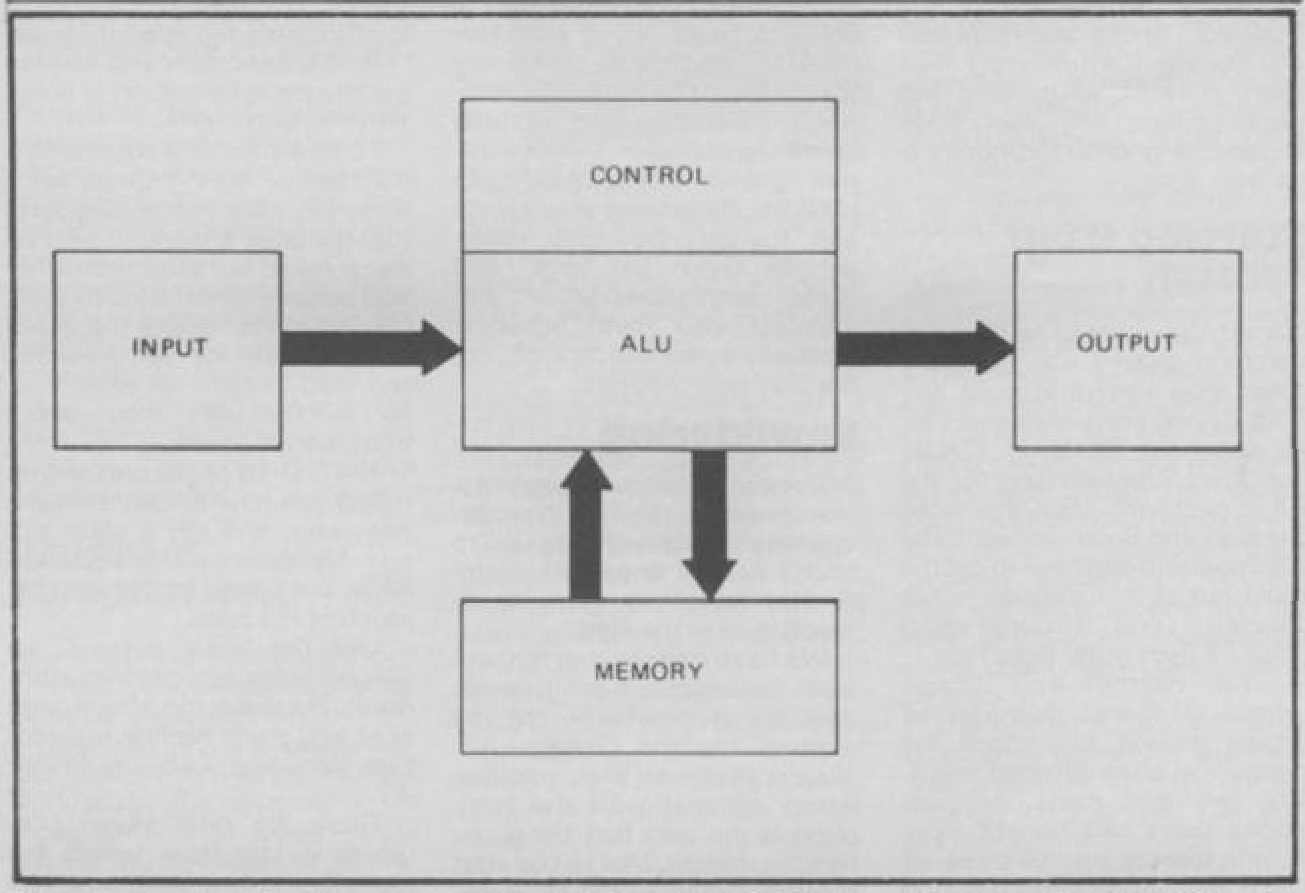
Programs must cater for the MSDs (and other peripherals) that they use. This can be a problem, due to the vast number of different devices around. A program designed for tape operation will not give its best (if it will work at all) when used with discs. A disc program will usually not work with tape. Disc systems by different manufacturers may require different commands.

Memory

Reading from, and writing to disc, while vastly faster than tape handling, can be exasperatingly slow. As memory becomes cheaper, an alternative is becoming available. The computer can be fitted with a great deal of random access memory (remember RAM?). For technical reasons, most small computers cannot normally use more than 64K of memory at a time (including the memory in which the BASIC interpreter is stored, and all user memory).

It is possible to get around this problem by making the machine look at one 'page' of memory at a time. Pages are smaller than 64K. Total memory then, can be anything you want. Values from 128K to 512K are common. It is possible to write a program which will make paged memory look like a disc to the computer. The only difference is that operation is almost instantaneous. You simply load the contents of the actual disc into the add-on memory, and then run the program entirely in memory. When you have finished, you store the whole contents of memory back on disc.





This diagram shows the five basic parts of every computer. On the left we have the INPUT to the computer, a keyboard, or a microphone, so you can speak directly to the computer or the computer can read the contents from a magnetic disc, magnetic tape or punched card. The heart of the computer is marked ALU here. This stands for ARITHMETIC LOGICAL UNIT. The ALU carries out the arithmetic computation and makes decisions. The CONTROL UNIT simply controls the flow of information throughout the computer. The MEMORY contains not only the intermediate and final result of any calculations being performed but also the information the computer needs to carry out the task you have given it, such as adding two numbers. The OUTPUT device can be any one of a number of things such as printed paper or the use of a TV-like screen. The CONTROL UNIT and the ARITHMETIC LOGICAL UNIT together make up what is called the CPU, the CENTRAL PROCESSING UNIT. Many modern microchips are a complete CPU, and input and output devices are added onto this chip, plus memory to turn the CPU into a microcomputer. (Caption and diagram are reproduced, with permission, from Tim Hartnell's book 'The Personal Computer Guide', published by Virgin Books.)

Spectrum rules the waves

Every publisher in the UK seems to have discovered the Spectrum, so there is sure to be a bumper crop of reading matter for Spectrum owners in the coming months.

Our review panel have been looking at a selection of the Spectrum books and finds that the standard is uniformly high (both in terms of content and presentation), with each book representing value for money in its own way.

Starting from scratch

Three titles seem to be aimed at common ground; the first-time user, who knows little (if any BASIC) and certainly knows little about the Spectrum. These are 'Easy Programming for the ZX Spectrum' (Shiva — Ian Stewart and Robin Jones); 'The ZX Spectrum and how to get the most out of it' (Granada — Ian Sinclair); and 'Programming Your ZX Spectrum' (Interface — — Tim Hartnell and Dilwyn Jones). Of course, they all tread similar ground, but attack the problem in quite different ways. We feel that many inexperienced users will benefit from buying two (or even all three) of the books, as what is unclear in one can be illuminated by the second or third. The funniest of the three is certainly Jones and Stewarts' 'Easy Programming...' which follows on the style of the highly successful 'PEEK, POKE, BYTE and RAM' title for the ZX81, and is enlivened by a batch of weedy cartoons.

There is no doubt, if you want your learning leavened with humour, that this is the title to start with.

The book goes through all the aspects of the Spectrum you're likely to need in the first few months, generally explaining them well and clearly. The demonstration programs include some remarkable demonstration graphics, ones which are ideal for impressing your family and friends. Our only reservations about the book, and these reservations should not distract you from what is generally a very fine product, are the first two chapters.

Bewildering

A somewhat obscure program is 'explained' in the first chapter, in a way which — unless you're a very careful or patient reader — may leave you more bewildered than in the light.

We have noticed that authors from academic backgrounds (like the man who wrote the manual for the Dragon 32) always believe that manipulating arithmetic on the computer is the very first thing you need to master. Most of us start by playing games. The exponention function we can live without for a while. Despite this, Stewart and Jones plunge into algebra, and the hierarchy of operations on pages eight and nine, when we imagine many readers really want to zap a few aliens.

Fortunately, things improve from this point, and chapter three's tour of the keyboard is the clearest of the three books we're considering.

Readable

The book continues to improve, in readability and interest as it continues, almost as though the authors were learning about the machine as they wrote the book, and by about page 30 were really starting to enjoy it.

When you're next in a bookstore, turn to page 117, where a blurred screen photo fails to do justice to a particularly fine graphics demonstration program, described as 'another offering of winsome beauty'. Clive Sinclair should buy the rights to this program, so he can run it every time a critic mouths 'But it's not true high resolution'.

The book includes 26 complete programs (described as 'prepackaged'), each of which will amply repay the time taken to enter them. In conclusion, 'Easy Programming...' is a book which will take you a long way into the mysteries of the Spectrum; is written with a consistent, and humourous hand; and shares the affection the authors feel for the computer.

For the Spectrum

Ian Sinclair's (no relation) book, 'The ZX Spectrum and how to get the most from it' is, in some ways a lighter book.

To fill a third of a page with a diagram of how to wire up a three-pin plug hardly suggests that the book is going to contain much meat, but after this rather odd beginning, Mr Sinclair gets into his stride. Unlike the Shiva book, which for some reason plumped to reset all programs, Mr Sinclair uses them direct from the ZX printer.

The first 18 pages are used up telling you how to connect up a television, and get a clear picture. Necessary information perhaps, but surely not worth that much of the book.

Like the Shiva authors, Mr Sinclair leaps straight into arithmetic, but drops this after a page or so and starts his first real section, on using TAB and PRINT AT.

There are only seven programs in this book which are more than a few lines long, and only one of them (ANIMALS) are you likely to run more than a few times.

Giving ideas

Against this is the fact that many of the sections in the book give a tremendous source of ideas for producing your own programs, and the chapter 'Roll Your Own', which explains how the idea of a program can be refined and built up, and then turned into a computer program, is worth the cost of the book alone.

We were, however, a little disconcerted to read in the introduction Mr Sinclair thanking Clive's PR people for 'the loan of a Spectrum' which suggests that the author had little time to build up real expertise on the machine before writing the book.

Have a look at the 'Roll Your Own' chapter (from page 69), 'Graphics' (page 81) and 'Sound Sense' (page 105) to see how well, and clearly, the author covers his material.

Tim Hartnell, the editor of this magazine, has been very industrious, and produced *two* books for the Spectrum already.

Although he is editor, he told us when presenting the books for review, 'be rude if you have to'.

And we've tried very hard to be!

The two books are 'Programming Your ZX Spectrum'; published by Interface, and 'The ZX Spectrum Explored', published by a company which Uncle C part owns, Sinclair-Browne. Clive has written the introduction to 'Explored'.

'Programming Your ZX Spectrum' is the thickest of the four books reviewed in this section, and also the most expensive. As a comparison, the Shiva book has 140 pages, Ian Sinclair's has 130, 'The ZX Spectrum Explored' has 218 and 'Programming Your ZX Spectrum', 230. The books each cost £5.95, except for 'Programming Your...' which is £6.95.

Now for the rude bit.

Tim, and his co-author Dilwyn Jones include many, many programs ('over 100 programs and routines' the publicity material says), but we felt many of them were introduced a little too early in the text, without adequate explanation. At the end, the techniques will be completely clear, but some readers will have to exercise patience and enter the programs on trust, knowing that the reasons for all the lines will become clear eventually.

Despite this, the book unfolds in a fairly logical way, and contains an easy-to-follow course on BASIC, and on programming, as well as on 'making the most of your Spectrum'. The program examples are, on the whole, good, with several 'major' programs including Reversi (Othello), Final Circuit and Life (with, for some reason, two completely different versions). A slight bug we found in Reversi has, we are assured, now been corrected in a reprint. The graphics chapter is

clear, and although some of the demonstrations are not as 'winsome' as those in the Shiva book, they still provide fodder for impressing the neighbours.

Graphics

The chapter on user-defined graphics is particularly clear, perhaps the clearest of all the books reviewed, and leads into a long, long 'Pacman-like' (very vaguely Pacman-like) game called Dotman.

This book, despite its headlong rush at the beginning, is certain to act as a resource which you'll take a long time to use up. At least one of the other books seems more likely to be 'exhausted' after the first two weeks of use.

Tim's other book, 'The ZX Spectrum Explored' is really a drawing together of the expertise of a number of people, including Jeremy Ruston, James Walsh and Tim Rogers.

It is *not* designed for the first time user, although a longish chapter at the beginning ('Programming in BASIC') has been included so you can make sense of the computer without any other of the guides.

Not all sections of this book are likely to appeal equally to all users, but no matter why you bought your Spectrum, you're probably going to be able to find material of real interest.



Expertise

When writing each section, Tim drew on the experience of an 'expert' in the field, so each section reflects a knowledge of that particular area which is likely to be more concentrated and detailed than Tim could possibly have written on his own. The section headings will show the kind of material covered: Programming in BASIC; Exploring the Spectrum's Colour; Exploring the Spectrum's Sound; Using the Spectrum in Business; Using the Spectrum in Education; Playing Games with your Spectrum; Three-dimensional graphics; Introduction to Machine Code; and A guide to Better Programming. The appendix includes a potted 'history of the computer' which moves very rapidly from Charles Babbage to some guy called Clive Sinclair. 'The race to produce ever-smaller, ever-more powerful computers was underway. Clive Sinclair entered that race in the late seventies'. One suspects that the Sinclair PR machine helped write the 'history'.

There are many, many programs (the business section even includes a very junior 'Visicalc'), with the best programs in the games section.

Take Advantage of your Spectrum's ROM, by David Reidy, is the first booklet looking at the insides of the Spectrum's ROM. It does not claim to be a complete book on the subject, as we have come to expect from Ian Logan, but is rather a taster of useful routines, where to find them and how to use them. It is fairly basic in its packaging and production (handtyped) but this is quite adequate for a quick reference booklet.

It covers such topics as where to put machine code on the Spectrum, and how to enter it, and then goes into some depth about several useful routines in the ROM.

Character display

Keypress detection is discussed, as well as a rather complicated section on displaying characters. We felt, here, that the author's rather complicated method of calculating DF CC was totally unnecessary. There are also various routines to use the BEEP facility, the printer and attributes file.

It is not an exhaustive coverage of ROM routines and their uses, but certainly provides a useful resource.

Program compilations

There are four books we'll look at which are collections of programs for the Spectrum.

The first, 'The Cambridge Colour Collection' is by Richard Altwasser, the designer of the Spectrum. The book costs £6.95 and contains just 20 programs. The games include 'Maze', 'Lunar Landing' and 'Android Nim', with more serious programs including 'Home Accounts' and 'Calendar'.

Andrew Hewson (who answers reader's problems in *Sinclair User*) has '20 Best Programs for the ZX Spectrum' for £7.95. The 'best programs' include 'Machine code editor', 'Index File' and 'Duckshoot'. Shiva has the cheapest book, the £2.50 'Computer Puzzles for Spectrum and ZX81', from the Stewart/Jones PEEK, POKE team. The many entertaining 'puzzles' include 'The wolf, the goat and the cabbage', 'Queens dominant' and 'The Magic Forest'.

David Harwood (who writes the 'Young Stuff' column in the users' club magazine Interface) has produced 'The Spectrum Software Library — 60 Games and Applications for the ZX Spectrum' for £4.95.

Although the emphasis is on games (including 'Checkers', 'Zap' and 'Galaxy Patrol'), the 'applications' include a 'Renumber' in BASIC and 'Chequebook'.



Publishers

Books reviewed in this section are published by the following:
Shiva Publishing Ltd., 4 Church Lane, Nantwich, Cheshire, CW5 5RQ.
Sinclair Browne Ltd., 10 Archway Close, London, N19 3TD.
Interface Publications, 44-46 Earls Court Road, London, W8 6EJ.
Granada Publishing, 36 Golden Square, London, W1R 4AH.
Hewson Consultants, 60A St Mary's Street, Wallingford, OX10 0EL.
Richard Altwasser, 22 Foxhollow, Bar Hill, Cambridge, CB3 8EP.
Hilderbay, 8/10 Parkway, Regents Park, London, NW1 7AA.

Machine Code and better basic by Ian Stewart and Robin Jones, Shiva Publishing Ltd.

This book is in many ways a sequel to Ian Stewart and Robin Jone's book 'PEEK, POKE, BYTE, & RAM' which enjoyed so much popularity as a beginner's guide to the ZX81. Here then is the what to do next, or 'What do I do now that I've added a 16K RAM pack'. Clearly Stewart and Jones' answer is 'learn how to improve your ability to program in BASIC and start on machine code programming'.

This book is part of Shiva's 'friendly micro' series, and as you'd expect, it is written in a friendly way designed to make you feel at ease even when rather complex matters are being discussed. From this point of view the book is generally a success. About half of it is on improving your BASIC and the latter half is on acquiring a grounding of knowledge of machine code. The first section covers topics such as using data structures in general, and such things as arrays, stacks, queues and 'trees' in particular.

Whilst the writing style is warm and friendly, the text may not always be clear to all who read it as their very next step after 'PEEK, POKE'. The section on arrays, for example, contains a host of useful information, but one cannot help thinking that the subject matter could have been presented in an even more simplified fashion. The section which follows it on searches, is surprisingly only three pages long and only deals with 'binary searches'. A more thorough simple guide to a variety of search methods might have been useful.

Introducing the reader to stacks next seemed a little out of place, although I'd acknowledge that other readers might feel differently. It is not entirely clear, I'd imagine, to someone without a fair amount of experience of computing what a stack might be used for — and the 'Tower of Hanoi' example Stewart and Jones give would not necessarily have enticed me to spend long on this section were I a beginner. Perhaps the fact that they are laying ground here for their introduction to machine code could have been made clearer.

The central section on structured programming is generally well presented and we are led through three longish 16K programs.



Machine code

The last half of the book on machine code is certainly a worthy effort, but I am not convinced that it succeded as a plain man's guide to learning machine code. The task is certainly daunting, but throwing the reader straight into HEXadecimal was not perhaps the best approach. Nor perhaps was introducing a fictional central processing unit 'which might have been built in the 1940's, but it wasn't'. Moreover, the first machine code the reader is introduced to is not machine code at all, but half way there in pseudo mnemonics — perhaps a little confusing to those who know a little about machine code already.

All in all, then a generally good followup to PEEK, POKE, if not quite as successful in accord with its aims as the earlier book. Undoubtedly useful for those who have gained some mastery of their ZX81 and want to go further, but like so many books for popular microcomputers, I would urge that this book be read alongside others aimed at a similar level.

MACHINE CODE AND BETTER BASIC COSTS £7.50 (and is, we feel, rather overpriced).

What can I do with 16K? By Roger Valentine



This is the follow-up to 'What can I do with 1K?' by the same author. Whereas that book's task was to tell you how you can achieve something worthwhile with only 1K of RAM, this book has the job of instructing us in how to get the most from the 16K we now have access to.

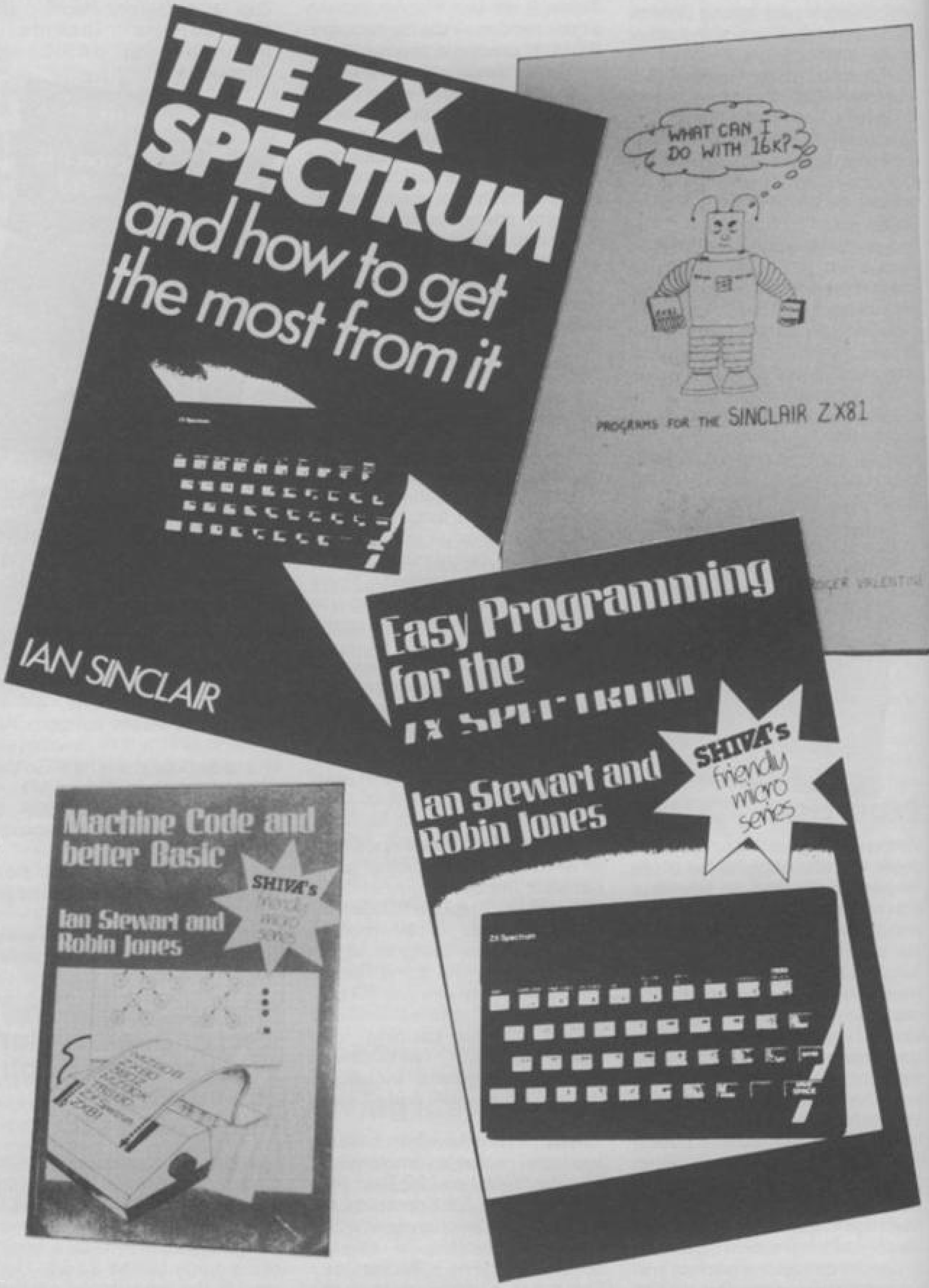
The entire book is in dot matrix print-out — probably with a CP80 or CP100, because there are no descenders. Nonetheless the text was clear and readable. The book shares much in common with the first half of Stewart and Jone's book, for it too aims to teach better programming techniques.

The book starts off with a program that may not be for everyone (a tarot card simulation), but is certainly an excellent program to learn the many techniques of structuring and memory conservation which Roger Valentine suggests. The rather more 'hands-on' approach of this book from it's beginning seems rather better than that of Stewart and Jones. That said I do not imagine the authors would be claiming to be teaching the same things. Other programs in this book make rather less explicit attempts to teach the reader any given fact about 'better programming', but rather give details as to how the program goes about its job. This is a pity, and perhaps Valentine might have given a little more discussion.

Confusing

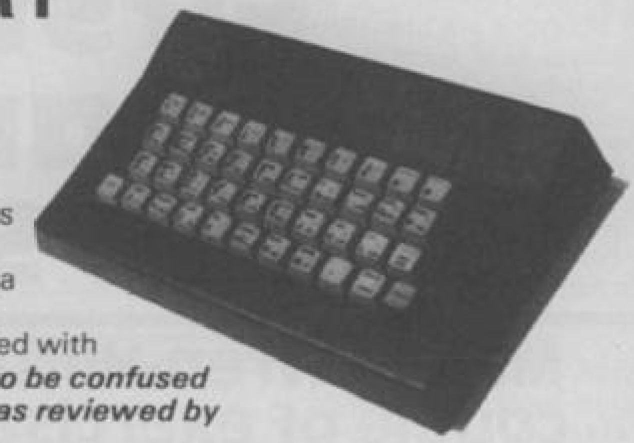
The lack of discussion certainly becomes apparent in the Battleships program, for the reader is asked to key in some machine code with no introduction as to what machine code is. Examples like this may make this book rather confusing for those readers with relatively little experience.

In summary, this book certainly shows you in a practical sense what you can do with 16K, but it may not be the best guide to how you can use 16K to best advantage. Nonetheless, many of the programs are excellent, and there are many useful hints for those who have programmed beyond Roger Valentine's What can I do with 1K? and books like it.



ZX81 & SPECTRUM KEYBOARDS COMPLETE WITH LUXURY CASE FOR ONLY

£37.95 & £45.00 inc VAT respectively



These highly professional keyboards incorporate executive buttons similar to those used on top-quality computers. A repeat key facilitates entering of characters much faster than is possible with a conventional keyboard, thus making it particularly ideal for cursor control and erasing. Attractive black plastic moulded case. Supplied with 12" flexible ribbon cable for connection into your computer. ***Not to be confused with toy keyboards currently on the market. Other attributes as reviewed by Tim Hartnell, ZX Computing/Interface.***

OTHER KAYDE HARDWARE/SOFTWARE

ZX81 4K GRAPHICS BOARD

The *KAYDE* Graphics Board is probably our best accessory yet. It fits neatly inside your ZX81. It comes complete with a pre-programmed 4K Graphics ROM. This will give nearly 450 extra graphics and with the inverse makes a total of nearly nine hundred.

The *KAYDE* Graphics Board has facilities for either 2K of RAM (for user definable graphics) 4K of ROM or our 4K Tool Kit Chips that will be available shortly. All the graphics are completely software controlled therefore they can be written into your progammes. Here are a few examples:

A full set of space Invaders — Puckman — Bullets, Bombs — Tanks — Laser Bases and Alien Ships

NO EXTRA POWER NEEDED

£19.95

16K GRAPHICS SOFTWARE

PECKMAN — The only true ZX version of the popular arcade game.

SPACE INVADERS — The best version available anywhere.

CENTIPEDE — ***"In all I think this is the best presented moving graphic programme I've yet seen" — Phil Garratt, INTERFACE***

£5.95 each

Graphic Software can only be used with a Graphics board

RAMPACKS . . . With a difference

Kayde Rampacks are exceptionally stable. They don't over-heat, lose memory, wobble or cause you to lose your program. They are fully compatible with all accessories and need no extra power, and therefore will run efficiently on existing Sinclair power supply. Simply plug them straight into the user port at the rear of your computer. Top quality massive add-on memory only:

£29.95 for 16K ZX81 Rampack and
£48.95 for 32K Spectrum Rampack

FLEXIBLE RAMPACK RIBBON CONNECTOR only £12.95

16K 81 SOFTWARE

3D/3D LABYRINTH — A cubit maze that has corridors which may go left, right, up and down. **£5.95**

4K Tool Kit full of utilities to aid the programmer in constructing and de-bugging E. PROM version for use with graphics ROM. **£9.95**

VIC 20 SOFTWARE

THE KAYDE VALLEY
OTHELLO
TWINKLE TWINKLE
KAYDE PECKMAN
APPOLO 8
SPACE ATTACK
PLUS MANY MORE
All £7.95 EACH

Vic 20 dealer

Overseas please add £2.50 p & p for all hardware and 75p for all software. All hardware comes fully built and tested and complete with a 14 day money back option. PRICES INCLUDE V.A.T.

SEND FOR A FREE CATALOGUE STATING TYPE OF COMPUTER.

FAST IMMEDIATE DELIVERY

KAYDE

KAYDE ELECTRONIC SYSTEMS LTD.

THE CONGE, GREAT YARMOUTH, NORFOLK NR30 1PJ
TEL: 0493 57867 TELEX: 975 247 CHATCOM G

TO: KAYDE ELECTRONIC SYSTEMS LTD,
DEPT.ZX12 THE CONGE, GREAT YARMOUTH,
NORFOLK NR30 1PJ

PLEASE PRINT 6

Qty.	Item	Item Price £	Total £

Please allow £1.50 P&P for all Hardware 50p for all software

VISA

DON'T FORGET YOUR CAN ALWAYS ORDER ON THE TELEPHONE WITH YOUR CREDIT CARD. DEALERS WELCOME.

*I enclose a cheque/postal order payable to *KAYDE Electronic Systems Ltd.*, for £ ....

*Please charge to my Access/Barclaycard/Trustcard account no.

* Please delete/complete as applicable.

Signature ____________________

Name: Mr/Mrs/Miss

Address

First steps in programming — on the Spectrum

Martin Wren-Hilton and Tim Hartnell outline a couple of exercises in programming, if you've never touched a computer before you bought your Spectrum.

Despite what you may have read in Spectrum's manual, this computer from Sinclair is actually quite easy to program. We have the feeling that the manual was written by a computer expert for other computer experts to read, what with all that rhubarb about Procrustean assignments and calculator stacks, it certainly seems complicated. But it is not all that difficult at all, as we shall see . . .



Setting up

First of all, it would be best to have a Spectrum in front of you. If you have not, then you should still be able to follow the article. Make sure that everything is set up, as shown on page 6 of the Introductory Manual. If you can, use a colour TV, otherwise a black and white set is better than none at all. Is everything set up? OK, now plug the little jack into the 9V DC socket on the back. Did the message "(C) 1982 Sinclair Research Ltd" come on? If not, then make sure that the television is correctly tuned in and that all the leads are in the correct place.

Touring The Keyboard

Have a good look around the keyboard and peel off the protective plastic if you have not done so already. Each key has about six functions — look at "Q" and you will see SIN in green, Q and PLOT in white, and ASN and a funny symbol that means 'less than or equal to' in red. Press ENTER and you will see the flashing "K" cursor appear. That "K" means that the computer is waiting for a keyword. The keywords are written in white on the key (like PLOT, DRAW and REM). Press the key marked PRINT — it's second down, on the right. You can see that the word PRINT has appeared on the screen and that the cursor has become an "L". This means that your Spectrum is now faithfully waiting for you to press a letter (or to change the cursor to a "G", "E" or "C"). Now is your opportunity to become famous and see your name on the goggle-box: type your name. If you make a mistake, press CAPS SHIFT and 0 — this will move the cursor back one, deleting the offending character.

Now press ENTER. When you press ENTER, the computer accepts what you have just typed and tries to do what you have told it to do. Hopefully, and probably to your disappointment, the computer will immediately report back to you "2 Variable not found, 0:1". That little exercise was to help you get used to the keyboard. As another exercise, press P then type in the whole alphabet. At the end, press ENTER twice and try again. You will find that the more often you try it, the faster you get.

Now Let's See The Spectrum Do Something Useful

At this point, it will be well worth doing two things — (i) Have a look over chapters 3, 4 and 5 of the Introductory Manual and (ii) go and get a few sheets of paper and a pen — you will see what those are for later. I hope that after doing (i), you realise that the Spectrum can either be used directly or programmed. If you use the computer directly, it does what you tell it to do, there and then. If you progam it however, it stores a long series of instructions one after the other, ready to be executed (RUN) later. Each program is made up from many "lines" — each line consisting of a line number and an instruction (or instructions) for the computer to store and execute one by one when the whole program is ready to RUN.

Writing a Program

Make sure that your Spectrum has nothing at all in its memory by pulling the power plug out of the back of the computer and re-inserting it, or getting the computer in keyword mode, where the cursor is a "K" and press NEW (on key A) followed by ENTER. To start writing a program, we must begin with a line number. A line number may be any number between 1 and 9999, and most people begin their programs at line 10. So type in 10. As you can see, the computer is still waiting for a keyword, so press the key L, as we want the word LET on the screen (and in the program). The cursor is now indicating that it wants a letter, so press A and you will see a little A appear. What we want now is the = sign on the L key, and in order to get it, we must press SYMBOL SHIFT and L. (If you are not used to using shift keys like those on typewriters, you must press SYMBOL SHIFT and whilst holding it down, press L, then release L before releasing SYMBOL SHIFT). Finally, press 1, followed by ENTER. This should appear at the top of the screen:

```
10 LET a=1
```

If it does not, then you can either EDIT the line by pressing CAPS SHIFT and 1 and moving the cursor backwards and forwards using SYMBOL SHIFT and either 5 or 8, deleting where necessary by pressing SYMBOL SHIFT and 0 or NEW the program as outlined above and start

again. Now enter the following lines obeying the rules that you have learnt so far:

```
20 PRINT a
30 LET a=a+1
40 IF a<>23 THEN GO TO 20
```

The only thing that I need to point out is that the < >in line 40 is one key W and should not be confused with the symbols on keys R and T.

Note that the line numbers go up in 10s. This is because at a later stage, you may wish to enter a program line between, say, 20 and 30, and call this in-between line 25. What does the program do? Press RUN then ENTER to find out. The screen looks like this:

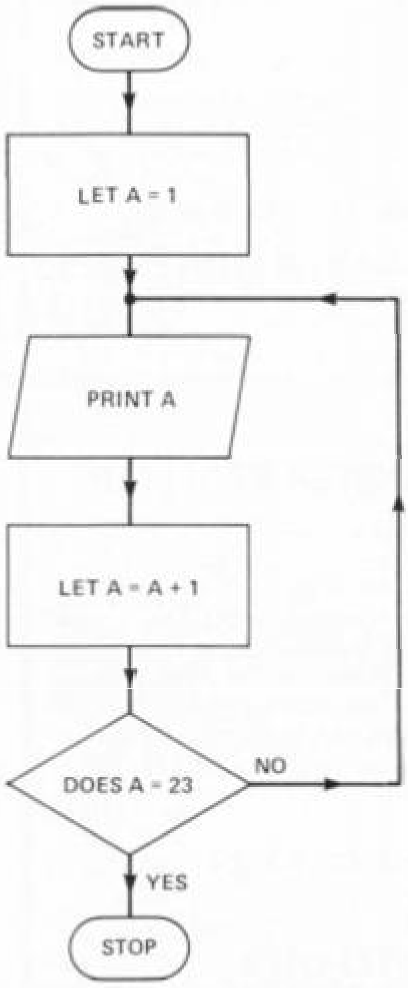
```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

0 OK, 40:1
```

So how did the program work? And precisely what did the computer do?

Variables

Starting at line 10, the computer set aside a variable called 'a' and gave that variable a value of 1. Line 20 PRINTs the value of the variable a on the screen (note: not the letter 'a'). In line 30, the computer gets the variable 'a', and increments it by adding one to it. The last line says "if 'a' does not equal 23, then go to line 20". We can represent the way that the program works in a flowchart, which looks like this:



Do not worry about the shapes of the boxes — they all represent program lines, except START and STOP, which the Spectrum does automatically when you RUN the program.

The Real Thing

From this point, let us move to a 'real' program. Enter the following,

```
 10 LET a=INT (RND*50) +1
 20 LET b=0
 30 LET b=b+1
 40 PRINT AT 3,3; INK 2;"Guess number ";b
 50 INPUT "Enter your guess ";c
 60 IF c=a THEN GO TO 110
 70 PRINT "No, that was wrong"
 80 PAUSE 50: CLS
 90 BEEP 1,50-ABS (a-c)
100 GO TO 30
110 PRINT "Yes, I was thinking of ";a
120 PRINT "It took you ";b;" guesses"
```

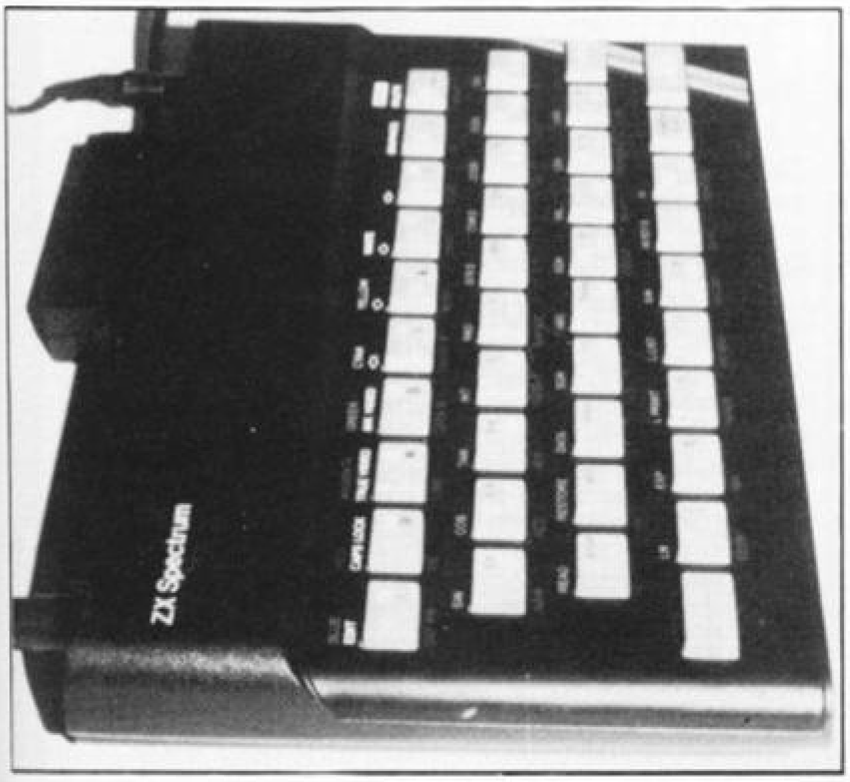
You get INT and RND in the first line by pressing both the CAPS SHIFT and the SYMBOL SHIFT keys at once, then pressing the keys where the needed word appears above it in green.

You get AT (in line 40) by holding down the SYMBOL SHIFT key and pressing the 1 key. The comma comes from the N key (you must hold down the SYMBOL SHIFT before you press it), and the semi-colon from the O key.

To get the word INK (which changes the colour in which words are printed), press both the SYMBOL SHIFT and the CAPS SHIFT at the same time, then release the white CAPS SHIFT, while continuing to hold down the SYMBOL SHIFT. Still holding down SYMBOL SHIFT, press the X key, and the word INK should appear. If it does not, go through the process again. Note that there is a colon (RED SHIFT, press the Z key) between two halves of line 80.

In this program, the computer thinks of a number from one to fifty, and you have to guess it. The feedback from the computer, on how good a guess you made, comes from the BEEP. The higher the note, the *closer you are to the correct* guess. You'll find you soon get quite skilled at working out what the computer's number should be. When you get it right, the computer will tell you how many goes it took you to guess it.

Once you've understood these two simple programs, you should be in a much stronger position to understand the other contents of this magazine, and to write your own programs.



On Your Marks

ON YOUR MARKS is a suite of educational programs for children aged six to 10. Written for the 16K ZX81, the programs aim to make the maximum use of the graphics capabilities of the computer.

Probably the most usual feature of this software is that it is a series of programs linked together and loaded as one. It uses every available byte of the 16K memory to do this. Why? The reason is that in a classroom — where it has been well tested — the teacher loads the whole suite before the class arrives. It takes more than six minutes but that doesn't matter. Once loaded the children can jump from one "game" to another at the press of a key. Program selection is from a menu, error trapped so that only the correct numbers have any effect.

One of the main criticisms heard from teachers about "drill" type computer programs is that they are boring. Computatutor put a lot of emphasis on visual presentation. The authors do not claim great originality for the content since all educational programs at this level comprise of setting questions and checking whether the answers are right or wrong. What is claimed is flair in presenting the material.

Packaging

The program once loaded begins to run automatically. It starts with animated graphics which are great fun to watch and then goes into a menu selection page. Pressing key one takes you to the first program available. You know you have got the right program because each one announces itself by name using an animated nest of boxes with the title appearing at the centre.

Find the key

The first program is *FIND THE KEY*. Its educational purpose is to help children become familiar with the keybaord. The computer randomly selects from either alphabetic or numeric keys and asks the child to press one. Any key pressed other than the correct one has no effect. When the correct one is pressed the child is told how many seconds it took them to find it. After ten successful attempts the times achieved for each key is displayed and the child is given the choice of trying another batch or returning to the menu.

Frogs

Two versions of FROGS are provided to test mental arithmetic at either multiplication or addition. The program is a development of one by Bob Maunder in his excellent 'The ZX81 Companion'. The idea is that every correct answer results in a man jumping over stepping stones towards his goal — a store of gold coins which turns to frogs as time passes. The thrill of this game is that it is played against a clock counting down. In the enhanced version, messages are flashed on the screen to step up the pressure as time passes. When all the answers are correctly answered the user is told how well they have done. On selecting FROGS the child is asked if they need instructions. Since the story is an essential part of the enjoyment it has to be told well by the software. If required the instructions are scrolled up the screen letter by letter in a very fascinating dispay.

Shoot the number

In a similar way, shoot the number has step-by-step instructions to explain what is expected of the user. The game presents a target practice in which the targets are a set of numbers. All the numbers are multiples of a number which appears on the butt of a moveable revolver. Pressing any number will fire the gun but the bullets are blanks unless the number pressed is the one which, multiplied by the one on the gun, is equal to the target. If the correct number is pressed and the position of the gun is opposite the target, a hit is recorded.

Completing the suite is a step-by-step guide to addition in which a sum is broken down into tens and units and the principle of carrying over to the next column is carefully demonstrated.

Anagram

To balance the arithmetic, side B of the cassette has a word-game. Anagram is played at three levels of difficulty. The more difficult, the longer the words which have their letters mixed up. Two children can play against each other. One enters the word while the other is not looking. The computer then presents it in a jumbled up version. Alternatively the computer will select a word from a store in its memory. If you give up just type "Help" to get the answer.

ON YOUR MARKS . . . is available from Computatutor, 3 Thalia Close, Greenwich, SE10 9NA priced £6.95.

James Walsh, our pint-sized education reviewer, the terror of his school in Loughton, takes a look in this issue at three educational packages, for the ZX81 and Spectrum.

In this edition I have been asked to look at three educational packages for the ZX81 and the Spectrum. The first one that I'll look at is CALPACs new Spectrum "JUNIOR EDUCATION" cassette which (I quote) "contains eight attractive, easy-to-use programs for the 7 to 11 age group, including Maths, English, Junior Science and History. Colour graphics and sounds are used extensively".

The other two packages are both for the ZX81 and are produced by ROSE CASSETTES. I use the word package intentionally, as both cassettes contain no less than 6 full length programs. The two packages that I will look at shortly are: JUNIOR ENGLISH 1 and G.C.E. "O" level FRNECH.

As I said earlier (if you were listening), I would first like to look at CALPAC's JUNIOR EDUCATIONAL cassette or the ZX Spectrum, so here goes: . . .

The cassette itself comes well packaged with a specially prepared leaflet of teachers notes. Though they are brief, they do give the teacher, parent or pupil for that matter, a reasonably good idea of what each program aims to do and hence whether they fit their requirements. My only suggestion is that some hints on loading could be included for the uninitiated user.

Drawing pictures

The first program is called DRAW. The idea is simply that you can guide a dot around the screen and draw pretty pictures with it. It is reasonably advanced as it allows you to control a number of funtions listed below: INK, PAPER, FLASH, BRIGHT, OVER, INVERSE, INVISIBLE, ON, OFF, SLOW, FAST, COPY and ENOUGH. Though this program holds very little educational value as such, it is a very good way for a newcomer to find his or her way around the keyboard, and generally feel that they are in control.

The second program is called "HOMEOPHONES", or in other words, words that sound the same but mean different things. You are given a sentence with a blank space and two or three alternative answers. You then type in the answer that you think is right. If you are correct then a graphically represented plant will grow and if you get to the end it will flower. If however you are wrong it makes a rasberry noise and goes onto the next question. Though the idea and application is good there is no variety, and I feel

that most kids of the prospective age-group would get bored with the same questions and only one special drawing at the end. Next on the tape is "SUB", which is a basic subtraction program. It deals with the subtraction of larger numbers from smaller numbers in a very understandable way. If you get a question right then you get a pleasent tone and the bridge gets a little longer. If you get it wrong then the program will carefully take you through the right way of doing it. At the end a tank will drive across the bridge and stop if it cannot get to the other side. A very thoughtful addition is the facility to choose how the answer should be explained, depending upon the way in which the pupil has been taught so far. This is a reasonably good program, but it gets rather boring after a while.

Nature studies

"FLOWER" is by far the best program on the cassette so far. It first draws an accurate diagram of a typical flower and all its parts. It then goes on to tell you what they all do and how. It also tells you where they all are and test you on their location. This program is ideal for 2nd year science/biology pre-'O' Level course candidates, as this is a major subject.

We now flip the cassette over and start looking at the second four programs.

Tables

The first program on side two is called, quite aptly, TABLES. You are allowed to specify a particular table from 2 to 12, or if you prefer then a variety up to a specified table, (3 to 12). It will then ask you 20 questions. If you get a question right then a bit more of your graphic train is drawn. If you get one wrong you hear another raspberry tone and the offending table is displayed. When you have finished your score is displayed and your train is shown driving through the countryside. This program is very good, ideal for Junior school and lower secondary school pupils.

Next we come to "COMP" – an English comprehension program. A sentence is given with a blank and you have to choose one of four words to fit in that space. The special value of this program is that it allows you to enter your own words and phrases. Again, if you get the answer right a bit more of the plant is drawn, and if you get it wrong it . . . you guessed it, makes a raspberry noise.

Chariots

"ROMANS" is a simple question and answer program on Roman history. If you get enough questions right the horse will gallop away with its chariot. Personally I found that this was reasonably advanced for a Junior school and so would be useful in Secondary school as well.

The final program on this cassette is called "SPELL", which is quite naturally a spelling program. The difference being that you type in the words and clues so that pupils can be tested. This way a large library of words can be built-up.

In all this is a good cassette. There may be certain programs that you will decide not to use, but this is mainly due to the incredible variety on this one cassette. I agree with the maximum age of 11 for the programs, except for FLOWERS and ROMANS, which may be useful in secondary schools.

Overall Ratings: 7/10 for Quality: 8/10 for value.

This cassette can be obtained for £5.50 from the address below:

CALPAC Computer Software, 108 Hermitage Woods Crescent, St. Johns, WOKING Surrey. GU21 1UF.

This is definately a worthwhile cassette for any junior school teaching using a SPECTRUM in class.

Junior English 1

Junior English is one of many educational packages from Rose Cassettes. These cassettes come quite well packaged, with loading hints on the inside of the cassette insert, but with very little information about the programs themselves. In computer circles it is not always necessary to include documentation, but for the product which is not only aimed at the computer owner but also at the school environment, I do feel that some introductory notes are helpful. CALPAC have done this simply by including an A4 size sheet folded into an A5 leaflet with a couple of sentences on each program. One small touch which I did like about the Rose Cassettes loading hints, was that they included approximate loading times for each program. This gives the user some idea of how many cups of coffee he or she has got time to drink before starting work!

Junior English is made up of six full 7K to 12K programs all of which are recorded once on the cassette. It may seem a better bet to record them twice, but reliability is almost 100%.

Meanings

The first two programs are called MEANINGS 1 and 2 respectively. The idea is that a phrase or word is displayed at the top of the screen, and underneath are three possible meanings. Your job is to decide which is the correct meaning and then type 1,2 or 3. The program holds a large stock of questions and answers so that you don't get the same questions on each RUN. The only difference between 1 and 2 is that 2 is slightly harder than 1.

The next program is called PARTS OF SPEECH. The idea is that you fill in the blank with one of three words displayed.

The first program on side two is called PROVERBS. It holds 35 well-known proverbs, such as: "A stitch in time saves nine". It displays one at a time but with a word missing so that you have to complete the proverb. I particularly liked this program because it gave a little variety in a subject which even with the best teacher in the World, most people find tedious after a while.

Learning English

We now come to "SIMILES", an aspect of English which is more popular (if that is the right word) than most. This time the program holds 40 well known similes. The program itself follows a very similar format to the rest, in that you have to fill the blank with one of the three possible answers.

The final program is called "ANAGRAMS". What you have to do is unscramble one of 50 six-letter words. If you cannot get the answer simply type 'Help' and the computer will display the first letter of the unscrambled word.

One particularly nice part of each program is the 'Jackpot' game which is played every time you answer a question. Though they are very basic, there is a different one on each program. A very important point when using a program in class is its' 'uncrashability', if you would pardon the expression. In other words, it is important that the computer checks every INPUT before it tries to digest it. This is very well done, as it is almost impossible to crash without hitting the 'BREAK' key.

Overall, this is a very well thought out program and well worth the expenditure of £4.50 (for those studying English at Junior School).

This and various other educational packages are available from: ROSE CASSETTES, 148 Widney Lane, Solihull, West Midlands, B91 3LH.

Rose Cassettes:

Summing it up

Ian Marshall of Bramhope, Leeds, sent us this program, with the following comments:

"Most of the teaching programs for simple maths seem to be of the type 13 + 14 = 27, where the calculation must be done mentally, (or with pencil and paper), and the answer entered in the sequence 'tens' followed by 'units'.

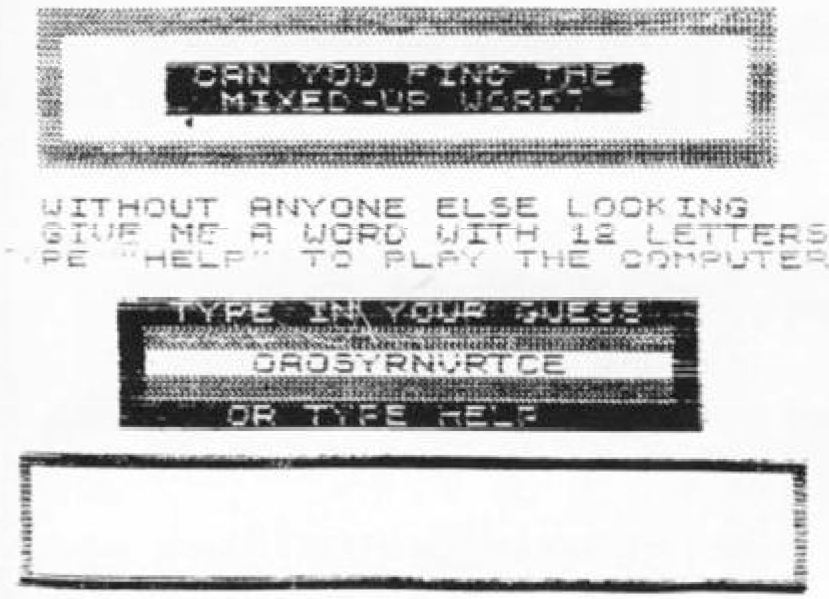
"In general, this is not the way a child is taught to add, and the facility to add the 'units' and then the 'tens' and enter the answer in sequence would be desirable.

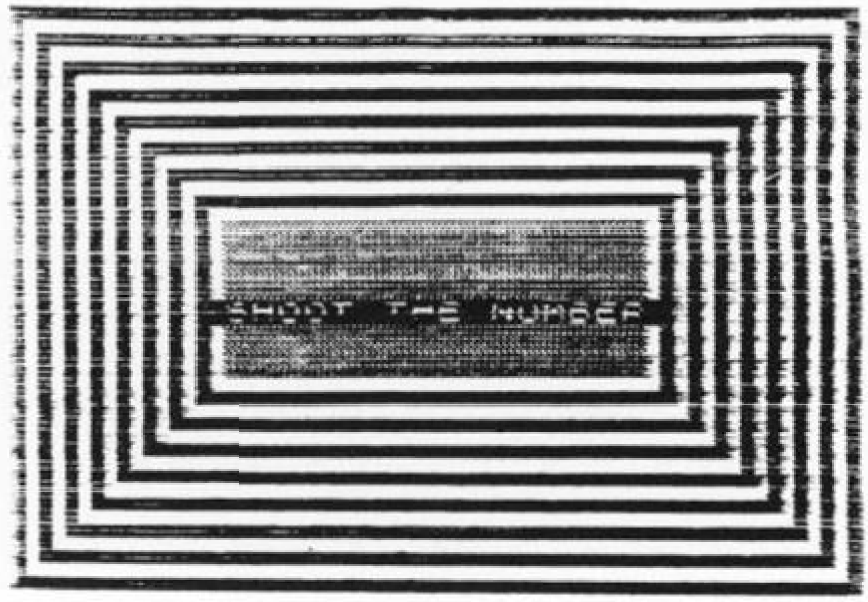
"The following program is very simple, but it follows what I believe is the desirable format and offers more sophisticated possibilities."

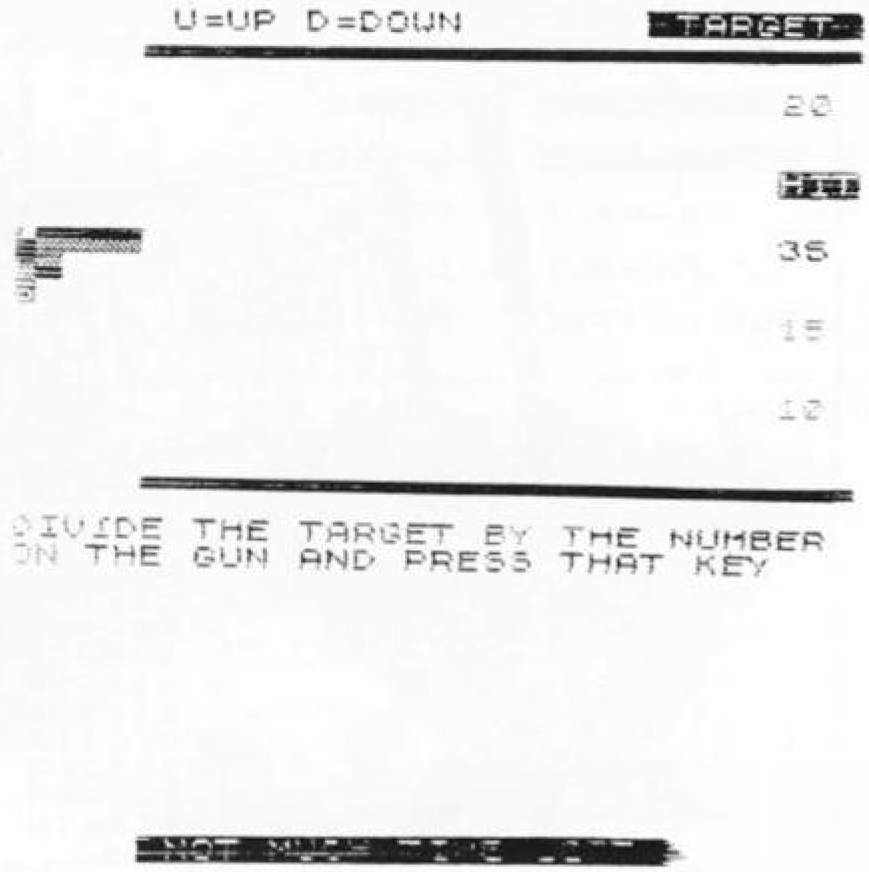


The headmistress of the Carlton and Netherfield Infants' and Nursery school in Nottingham, says she believes it is important for her pupils to familiarise themselves with technological developments at an early age, so she's taken her ZX81 along to school. Miss Johnson, shown in the photograph with Ian Preston at work on the computer, says the children use it for multiplication and division, and for spelling games.

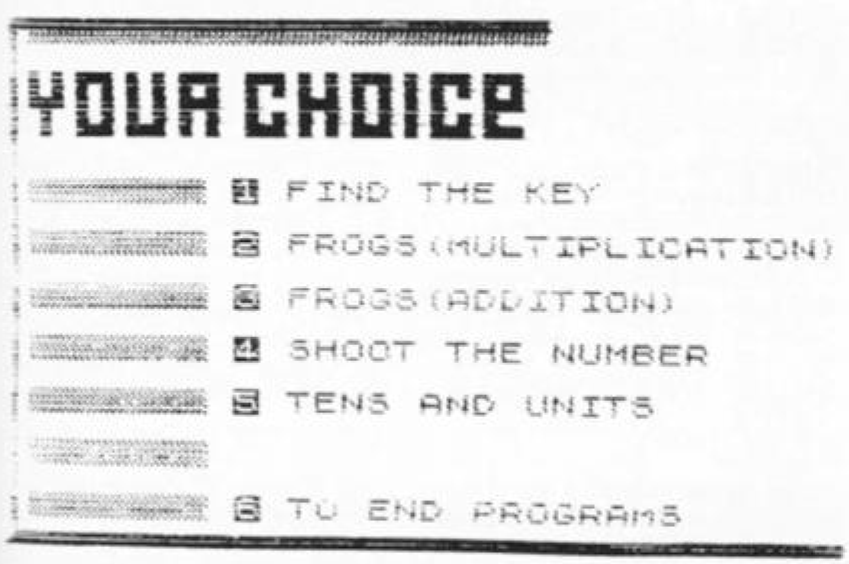
"The children play number and spelling games, and see their names printed on the screen. It is important that they know there is more to computers than playing games." (Photo courtesy of the Nottingham Evening Post).

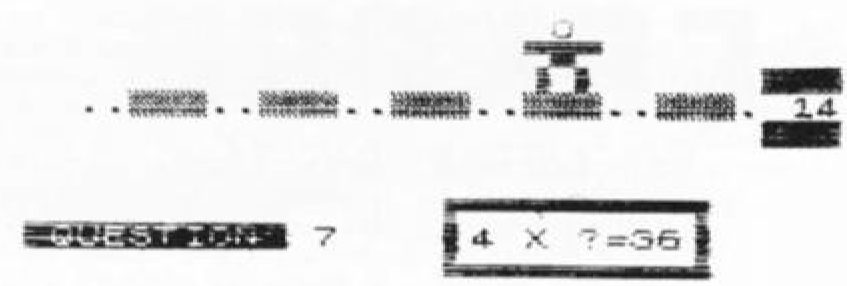

CAN YOU FIND THE
MIXED-UP WORD?
WITHOUT ANYONE ELSE LOOKING
GIVE ME A WORD WITH 12 LETTERS
TYPE "HELP" TO PLAY THE COMPUTER
TYPE IN YOUR GUESS
GROSYRNURTCE
OR TYPE HELP

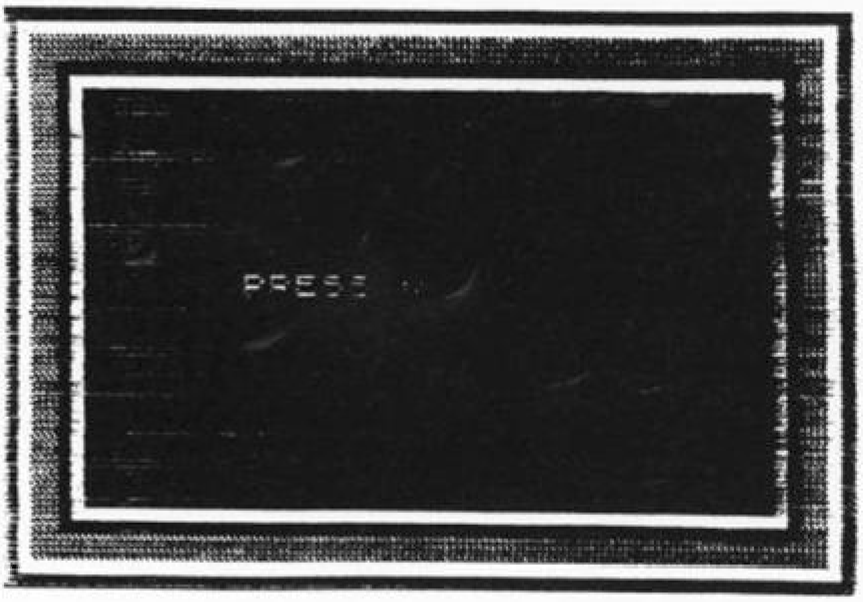



U=UP D=DOWN
TARGET
20
35
15
12
DIVIDE THE TARGET BY THE NUMBER
ON THE GUN AND PRESS THAT KEY


LEARNING
IS FUN
COMPUTATUTOR


YOUR CHOICE
1 FIND THE KEY
2 FROGS (MULTIPLICATION)
3 FROGS (ADDITION)
4 SHOOT THE NUMBER
5 TENS AND UNITS
6 TO END PROGRAMS


14
7
4 X 7=36




```
{                                    }
{   *** CERTIFICATE ***              }
{                                    }
{   NAME -       JAMES               }
{   DATE -       25.9.82             }
{   PROGRAM -    "SIMILES"           }
{                                    }
{      CORRECT: - 29                 }
{    INCORRECT: - 1                  }
{   PERCENTAGE: - 96                 }
{                                    }
```




```
Now switch off your tape
recorder.

Press ENTER to continue.
```

CALPAC

Educational Software

Picture plotter.

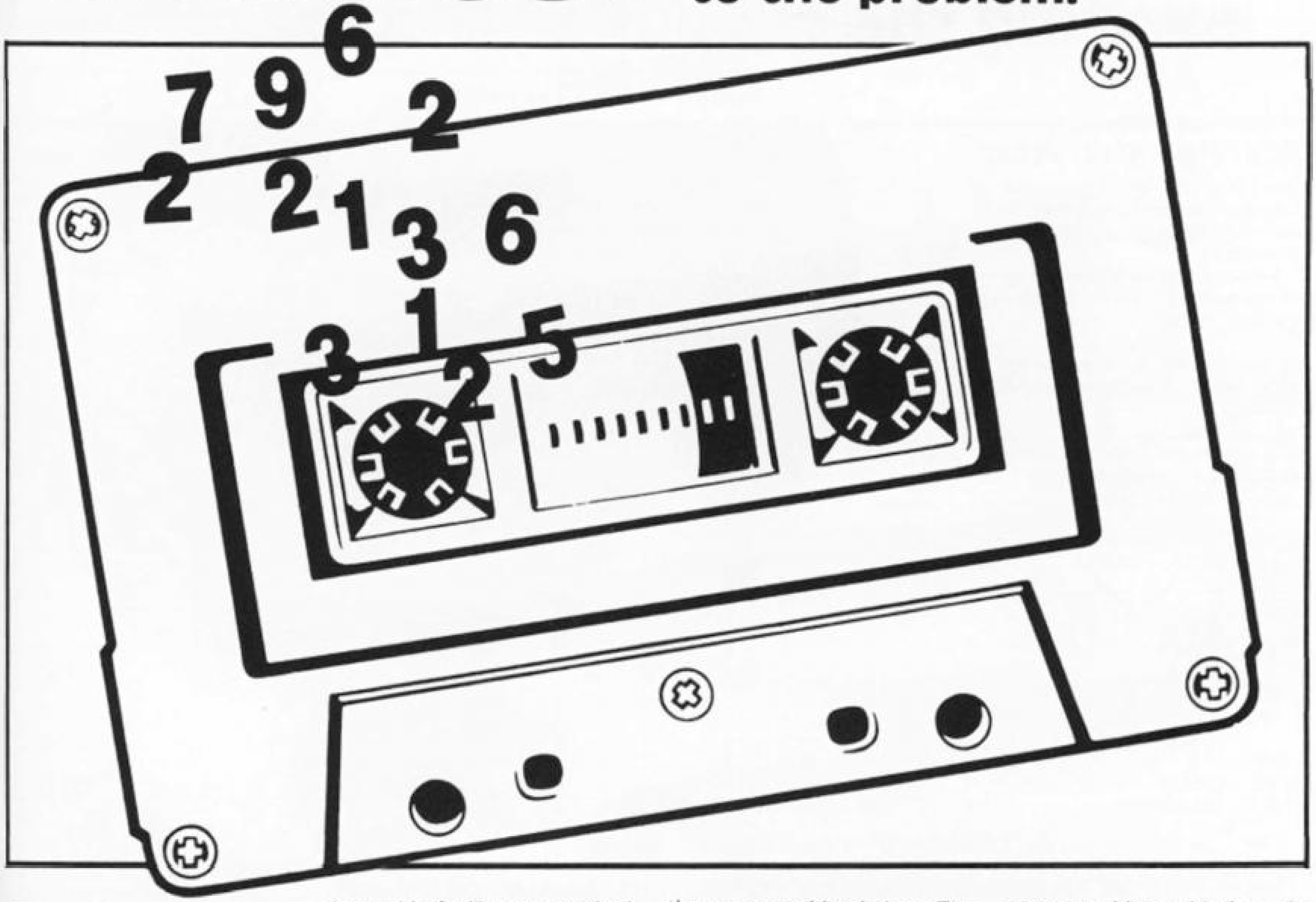
© 1982   J J Warren

Press e for an explanation.
Press ENTER to continue

```
  10 PRINT "MATHEMATICS FOR GENIUSES"
  20 PRINT
  30 PRINT "TRY THE SUM THAT FOLLOWS IN A"
  35 PRINT "FEW SECONDS,-THEY/RE ALL SIMPLE"
  40 PRINT "ADDITION SUMS - PUT THE UNITS"
  45 PRINT " IN FIRST, THEN THE TENS"
  50 PRINT "OK? AWAY YOU GO"
  60 PAUSE 800
  70 POKE 16437,255
  90 CLS
 100 RAND
 101 LET A=0
 102 LET B=0
 103 LET C=0
 110 LET A=INT (RND*19)+1
 120 LET B=INT (RND*19)+1
 130 LET C=A+B
 200 IF A<=9 THEN PRINT AT 10,6;A
 210 IF A>9 THEN PRINT AT 10,5;A
 220 IF B<=9 THEN PRINT AT 11,6;B
 230 IF B>9 THEN PRINT AT 11,5;B
 240 PRINT AT 11,7;"+"
 250 PRINT AT 12,5;"--"
 300 PAUSE 10E3
 310 POKE 16437,255
 320 LET D$=INKEY$
 321 IF INKEY$=CHR$ 118 THEN GOTO 300
 325 LET D=VAL D$
 330 PRINT AT 13,6;D
 340 IF D=C THEN GOTO 2000
 350 PRINT AT 13,8;"DON/T FORGET TO CARRY"
 360 PRINT AT 14,8;"ANY ""TENS"""
 400 PAUSE 10E3
 410 POKE 16437,255
 420 LET E$=INKEY$
 425 IF INKEY$=CHR$ 118 THEN GOTO 400
 430 LET E=VAL E$
 440 PRINT AT 13,5;E
 450 PRINT AT 13,8;"                    "
 460 PRINT AT 14,8;"                    "
 500 IF D=C THEN GOTO 2000
 510 IF C=D+(E*10) THEN GOTO 2000
 520 IF C<>D AND C<>D+(E*10) THEN GOTO 5000
2000 PRINT AT 13,8;"WELL DONE"
2010 PRINT AT 14,8;"-ANOTHER GO? (Y/N)"
2015 PAUSE 10000
2016 POKE 16437,255
2020 IF INKEY$="Y" THEN GOTO 90
2030 IF INKEY$="N" THEN GOTO 6000
5000 PRINT AT 13,8;"OH DEAR-WRONG"
5010 PRINT AT 14,8;"LET/S TRY THE SAME "
5020 PRINT AT 15,8;"SUM AGAIN"
5030 PAUSE 130
5035 CLS
5040 PRINT AT 13,5;"  "
5050 GOTO 200
6000 CLS
6005 PRINT AT 9,10;"OK MATHEMATICIAN"
6010 PRINT AT 10,10;"SO LONG FOR NOW"
6020 STOP
6050 SAVE "MATHS"
6060 GOTO 10
```

Adding a tape counter

It can be difficult to find programs on cassette when you have a number of them on the same cassette, and no tape counter on your recorder. Nick Pearce took the D.I.Y. approach to the problem.



On a shoestring

Computer hobbyists who, like I have to, operate on a shoesting budget, might be interested to read about a tape counter I adapted to fit my old Philips cassette recorder. In common with other ZX 81 users, I find that only good quality audio cassettes give consistently good LOAD/SAVE results and in the interests of economy I save ten or so programs each side which makes a tape counter an essential piece of equipment. Some makes of recorder, including my Philips, do not have this facility as standard. My simple adaptation should fit most recorders.

It is basically a counter for a tape recorder which I purchased from 'Scoops' (90p) and which is adapted to be operated by the drive of the cassette recorder. The only permanent modification I made to the cassette recorder is a $\frac{1}{2}$-inch diameter hole cut in the lid.

The tape counter is mounted on a block of wood and rests on the lid of the cassette recorder. A piece of rubber tube fixed to the counter drive pully passes through the hole in the lid to make contact with the cassette drive below. The tube is flexible rubber, 8mm outside diameter, 5mm inside diameter, and must be a tube and not solid since it must bear on the annulus which drives the cassette and not on the central stationary stem. I used 'araldite' to connect the tube to the pulley of the counter, and a little care is needed to ensure that it is fixed centrally, although the rubber tube will take up small eccentricities.

Getting tricky

The only tricky part of the operation is to make sure the rubber makes contact with the cassette drive with just the right amount of pressure — too much and friction would slow the cassette down. I did this by making the tube about $\frac{1}{8}$ inch too long and then gluing thin rubber pads on the underside of the wooden block until I got the pressure right.

I find this adaptation a great help for locating programs on cassette. It does not affect the operation of the recorder; programs I SAVED before making it, LOAD successfully with it in position. Cheap too, 90p and half an hours work instead of buying a new cassette recorder with a built in tape counter!

DAZZLING DISPLAYS

Richard Wright shares a machine code routine to clear the ZX 81 screen of any character you specify, and Paul Holmes shows us just how exciting the output of the Spectrum can be.

Clearing a Character

The following short machine code routine clears the screen of any occurrence of a character you specify.

You specify the character by poking address 16507 with its character code.

e.g. for a £ sign you would enter as a direct command POKE 16507,12.

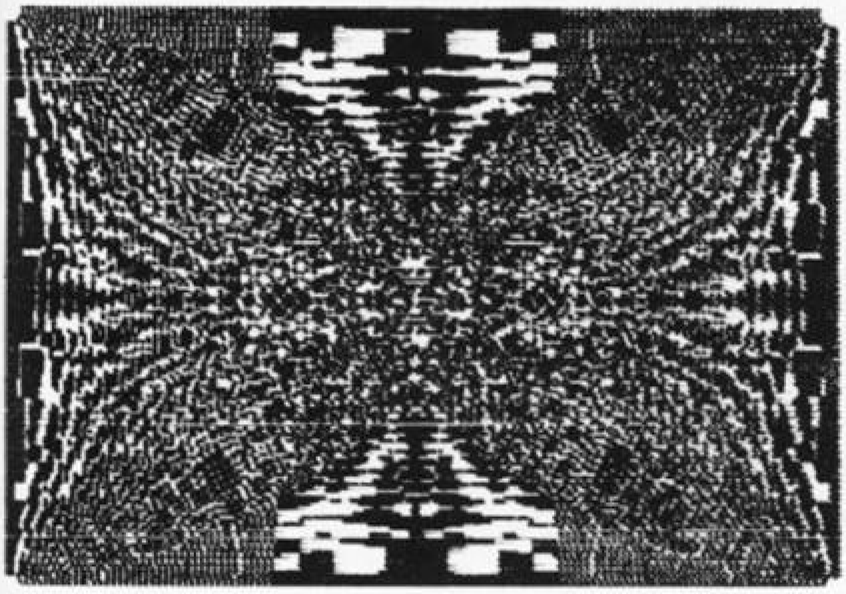
To enter the machine code enter the following program:

```
1    REM(33 characters)
10   LET X=16514
20   LET A$E" "
30   IF A$="" THEN IN-
     PUT A$
40   IF A$="S" THEN
     STOP
50   POKE X,16*CODE
     A$+CODE
     A$(2)-476
60   LET X=X+'1
70   LET A$=A$(3 TO)
80   GOTO 30
```

MNEMONIC	HEX CODE
LD B,22	06 16
LD A,(16507)	3A 7B 40
LD D,A	57
LD HL,(D-FILE)	2A OC 40
DEC HL	2B
INC HL	23
LD A,(HL)	7E
CP D	BA
JR Z,6	28 06
CP 118	FE 76
JR Z,7	28 07
JR,-11	18 F5
LD A,O	3E 00
LD(HL),A	77
JR,-16	18 FO
DEC B	05
LD A,B	78
CP O	FE 00
RET Z	C8
JR,-23	18 E9

Enter the codes under the "HEX CODE" column. The routine is called RAND USR 16514.







These programs will keep your Spectrum happy for hours at a time. In the first one, Test Card, the coarse checked pattern in line 150 should be entered as graphic 'b', and the fine checked pattern on either side is graphic 'a'. In the second program, Graphics Alive-oh, just enter the program — and stand back and watch.

```
  10 INK 6: OVER 0: PAPER 0: BOR
DER 0: CLS
  20 FOR i=-PI TO PI STEP .1
  30 PLOT 128,0: DRAW 0,175,i
  40 NEXT i
  50 PAUSE 400
  55 PAPER 3: OVER 1
  57 FOR j=0 TO 1
  60 FOR i=0 TO 255 STEP 3
  70 PLOT 0,0: DRAW i,175
  80 NEXT i
  90 FOR i=175 TO 0 STEP -3
 100 PLOT 0,0: DRAW 255,i
 110 NEXT i
 120 IF j=0 THEN PAUSE 200: BORD
ER 7: PAPER 7: INK 1: CLS
 130 NEXT j
 140 PAUSE 400
 270 BORDER 0: PAPER 0: INK 4: C
LS
 280 FOR i=-87 TO 88
 290 PLOT 127,87: DRAW -127,i
 300 PLOT 127,87: DRAW 128,i
 310 NEXT i
 320 FOR i=-127 TO 128
 330 PLOT 127,87: DRAW i,-87
 340 PLOT 127,87: DRAW i,88
 350 NEXT i
 360 PAUSE 800
 370 PAPER 6: INK 2: BORDER 6: C
LS
 380 FOR i=6 TO 171 STEP 2
 390 PLOT 0,i: DRAW i,-i,-PI/2
 400 PLOT 255,i: DRAW -i,-i,PI/2
 410 PLOT i,175: DRAW -i,-i,-PI/
6
 420 PLOT 255,175-i: DRAW -i,i,-
PI/2
 430 NEXT i
 440 GO TO 440
```

```
------------Test Card-----------

  30 FOR i=0 TO 7
  40 POKE USR "c"+i,1
  50 NEXT i
  60 FOR i=0 TO 4 STEP 4
  65 POKE USR "b"+i,BIN 110011
  66 POKE USR "b"+i+1,BIN 110011
  70 POKE USR "b"+i+2,BIN 110011
00
  71 POKE USR "b"+i+3,BIN 110011
00
  80 NEXT i
  90 FOR i=0 TO 6 STEP 2
 100 POKE USR "a"+i,BIN 01010101
: POKE USR "a"+i+1,BIN 10101010
 110 NEXT i
 120 INK 7: BORDER 0: PAPER 0: C
LS
 130 FOR i=1 TO 7
 140 FOR j=1 TO 3
 150 PRINT TAB 8; BRIGHT 0; INK
i;"████"; INK 7; BRIGHT INT (i/4
);"▒▒▒▒▒▒▒▒"; INK i; BRIGHT 1;"█
███"
 160 NEXT j: NEXT i
 170 INK 7
 205 FOR i=0 TO 62 STEP 2
 210 PLOT i,8
 212 IF i<33 THEN DRAW BRIGHT 1;
0,167
 215 IF i>33 THEN DRAW BRIGHT 0;
0,167
 220 NEXT i
 230 FOR i=222 TO 254 STEP 2
 240 PLOT i,8: DRAW BRIGHT 1;0,1
67
 250 NEXT i
 260 FOR i=8 TO 174 STEP 2
 270 PLOT 192,i: DRAW BRIGHT 1;6
3,0
 280 NEXT i
 290 PRINT AT 21,0;"Brt Nor Nor 
G ) 2G G Brt Lin Crss"
 300 PAUSE 1200: BORDER 7: PAUSE
 1200: BORDER 0: GO TO 300
9998 STOP
9999 PAPER 7: INK 1: BRIGHT 0: B
ORDER 7
```

Don't let its size fool you.

If anything NewBrain is like the Tardis.

It may look small on the outside, but inside there's an awful lot going on.

It's got the kind of features you'd expect from one of the really big business micros, but at a price of £269.95 including VAT it won't give you any sleepless nights.

However, let the facts speak for themselves.

You get what you don't pay for.

NewBrain comes with 24K ROM and 32K RAM, most competitors expect you to make do with 16K RAM.

What's more you can expand all the way up to 2 Mbytes, a figure that wouldn't look out of place on a machine costing ten times as much.

We've also given you the choice of 256, 320, 512 and 640 x 250 screen resolution, whereas most only offer a maximum of 256 x 192.

Big enough for your business.

Although NewBrain is as easy as ABC to use (and child's-play to learn to use) this doesn't mean it's a toy.

Far from it.

It comes with ENHANCED ANSI BASIC, which should give you plenty to get your teeth into.

And it'll also take CP/M® so it speaks the same language as all the big business micros, and feels perfectly at home with their software.

NO OTHER MICRO HAS THIS MUCH POWER IN THIS MUCH SIZE FOR THIS MUCH MONEY.



So as a business machine it really comes into its own.

The video allows 40 or 80 characters per line with 25 or 30 lines per page, giving a very professional 2000 or 2400 characters display in all on TV and/or monitor. And the keyboard is full-sized so even if you're all fingers and thumbs you'll still be able to get to grips with NewBrain's excellent editing capabilities.

When it comes to business graphics, things couldn't be easier. With software capabilities that can handle graphs, charts and computer drawings you'll soon be up to things that used to be strictly for the big league.

Answers a growing need.

Although NewBrain, with its optional onboard display, is a truly portable micro, that doesn't stop it becoming the basis of a very powerful system.

The Store Expansion Modules come in packages containing 64K, 128K, 256K or 512K of RAM. So, hook up four of the 512K modules to your machine and you've got 2 Mbytes to play with. Another feature that'll come as a surprise are the two onboard V24 interfaces.

With the aid of the multiple V24 module this allows you to run up to 32 machines at once, all on the same peripherals, saving you a fortune on extras.

The range of peripherals on offer include dot matrix and daisy wheel printers, 9", 12" and 24" monitors plus 5¼" floppy disk drives (100 Kbytes and 1 Mbyte) and 5¼" Winchester drive (6-18 Mbytes).

As we said, this isn't a toy.

It doesn't stop here.

Here are a couple of extras that deserve a special mention.

The first, the Battery Module, means you won't be tied to a 13 amp socket. And, even more importantly, it means you don't have to worry about mains fluctuations wreaking havoc with your programs.

The ROM buffer module gives you a freedom of another sort.

Freedom to expand in a big way. It gives you additional ROM slots, for system software upgrades such as the Z80 Assembler and COMAL, 2 additional V24 ports, analogue ports and parallel ports.

From now on the sky's the limit.

Software that's hard to beat.

A lot of features you'd expect to find on software are actually built into NewBrain so you don't need to worry about screen editing, maths, BASIC and graphics.

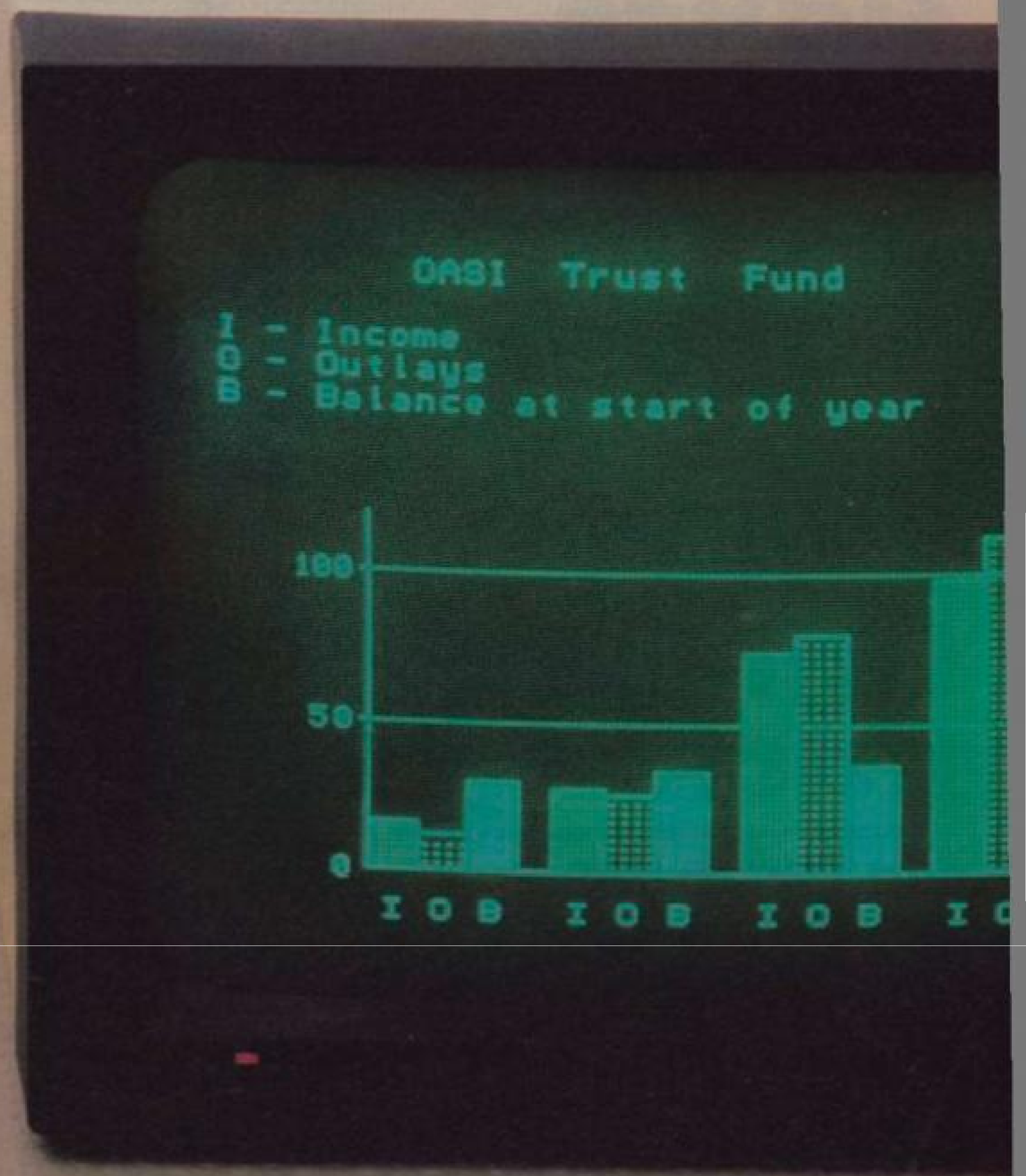
However, if you're feeling practical you can always tackle household management, statistics and educational packages. And because NewBrain isn't all work and no play, there's the usual range of mind-bending games to while away spare time.

Waste no more time.

To get hold of NewBrain you need go no further than the coupon at the bottom of the page.

With your order we'll include a hefty instruction manual so you'll know where to start, and a list of peripherals, expansion modules, and software so you'll know where to go next.

® CP/M is the registered trade mark of Digital Research Inc.



Technical Specifications

The NewBrain is a fully specified professional computer built to the highest standards of engineering and reliability. Chosen by leading OEM suppliers. Designed to facilitate easy expansion for use with the CP/M operating system, and the addition of 5¼ flexible and Winchester disks, 12 green phosphor professional standard monitor, 80 cps professional quality dot matrix printer with pin addressable graphics.

Z80A cpu and COP 420M input/output microprocessors. 32K RAM expandable to 2 Mbytes. 28K ROM. Dual Cassette Ports. UHF TV port. CCITT Monitor Port Video 40/80 Character x 25/30 lines. 256, 320, 512, 640 x 250 Pixels. Expansion Port. V24 Bi-directional Port. V24 Printer Port. 16-character display (AD only).

Software: Enhanced BASIC (ANSI x 3.2/78) Independent Operating System (12 device drivers). Multi Page Screen Editor (32 Control Commands). Maths (10 Significant Figures). Graphics (Absolute & Relative Plotting, Line & Arc Drawing, Shading, 20 English Language Commands).

NewBrain,
Grundy Business Systems Ltd., Grundy House,
Somerset Road, Teddington TW11 8TD.

To: NewBrain, Grundy Business Systems Limited, Grundy House, Somerset Road, Teddington TW11 8TD.

Please send me a full-colour leaflet, price list, and name and address of my nearest stockist. ☐

Name____________________

Address____________________

Postcode____________________

NEWBRAIN

ZX80 fights back

ZX80 fights back

ZX80 fights back



Ian Turtle has contributed two fine ZX80 programs, which prove you can still do a lot with the pilot of the Sinclair fleet. The third great ZX80 program comes from Andrew Haslam.

Klingon capture

A Klingon ship is hiding behind one of 100 stars. You must try and trap the Klingon by shooting the eight surrounding squares, without hitting the Klingon. He must be taken alive.

The display tells you where the Klingon was one move ago. This may or may not be his present position, as he could have moved.

The Klingon decides to move in one of nine ways. He may move to one of the eight surrounding squares, or he may stop still. The probabilities of these actions are equal. If the square it wishes to move to is clear, then it will move there, and the square it vacates will be shown as the 'last position' on the display.

If the square it wishes to move to has already been blasted away, it will remain in the same square.

When shooting, enter the horizontal co-ordinate, followed by the vertical one.

Klingon Capture Listing

```
10 DIM A(99)
20 LET B=RND(10)-1
30 LET C=RND(10)-1
40 LET A(B*10+C)=2
50 FOR G=1 TO 20
60 PRINT "±";
70 FOR I=0 TO 9
80 PRINT I;
90 NEXT I
100 PRINT
110 FOR I=0 TO 9
120 PRINT I;
130 FOR J=0 TO 9
140 PRINT CHR$(20-7*(A(I*10+J)=1));
150 NEXT J
160 PRINT
170 NEXT I
180 PRINT "LAST POS±";C;",";B
190 LET D=RND(3)-2+B
200 LET D=D+(D>9)-(D<0)
210 LET E=RND(3)-2+C
220 LET E=E+(E>9)-(E<0)
230 IF A(D*10+E)=1 THEN GOTO 260
240 LET B=D
250 LET C=E
260 PRINT "SHOT?"
270 INPUT N
280 INPUT M
290 IF M=B AND N=C THEN GOTO 440
300 LET A(M*10+N)=1
310 FOR I=-1 TO 1
320 FOR J=-1 TO 1
330 IF B+I<0 OR B+I>9 OR C+J<0 OR C+J<9 I=0 AND
    J=0 THEN GOTO 360
340 IF A((B+I)*10+C+J)=1 THEN GOTO 360
350 GOTO 400
360 NEXT J
370 NEXT I
380 PRINT "YOU WIN"
390 STOP
400 CLS
```

```
410 NEXT G
420 PRINT "YOU LOSE"
430 STOP
440 PRINT "DUMMY-YOU SHOT THE KLINGON"
450 STOP
```

Snooker

This game of snooker is a two player game. For those unfamiliar with snooker, we'll explain. The game consists of potting the balls. At the start of each break, you try and pot a red. If successful, you gain one point, and the right to aim at potting a 'colour', i.e. a non-red ball.

If you are successful again, you'll gain the number of points the ball is worth. Once you've potted a 'colour', you must try for a red again and so on. This continues until you fail to pot a ball, and the play then passes to your opponent.

The point values for the balls are as follows:

1 – red
2 – yellow
3 – green
4 – brown
5 – blue
6 – pink
7 – black

It is good policy to try and sink the higher value balls after each red, if you can.

Fouls

Before any shot, a table is displayed, giving the ball number and the chance of potting it, from one to ten (that is, impossible to fairly easy). If any ball has been previously potted, then a P will be shown against it.

Trying to pot an already potted ball will result in a FOUL being called, and four points will be given to your opponent.

Once a red has been potted, a similar table will be given for the other colours. You'll see that it becomes easier to continue with a break once the first ball has been potted.

When all the reds have been cleared, and a colour has been taken with the last red, the colours must be taken in the order 2, 3, 4, 5, 6, and 7 (that is, yellow, green, brown, blue, pink and black). If they are taken out of order, a FOUL is called.

Snooker Listing

```
10 LET Z = -1
20 LET W = 0
30 LET V = 2
40 LET X = 0
50 LET Y = 0
60 DIM S(1)
70 DIM C(7)
80 DIM R(15)
90 LET Z = Z + 1
100 IF Z>1 THEN LET Z = 0
110 LET B = 0
120 GOSUB 650
130 IF X = 15 THEN LET Y = 1
140 IF Y = 1 THEN GOTO 390
150 PRINT "REDS"
160 FOR I = 1 TO 15
170 PRINT I;"-";
180 IF R(I) = -1 THEN GOTO 220
190 LET R(I) = RND(10)
200 PRINT R(I)
210 GOTO 230
220 PRINT "P"
230 NEXT I
240 PRINT "POTT?"
250 INPUT P
260 IF NOT R(P) = -1 THEN GOTO 320
270 PRINT "FOUL"
280 LET S((Z = 1) + 1) = S((Z = 1) + 1) + 4
290 INPUT A$
300 CLS
310 GOTO 90
320 IF RND(R(P))>2 - 3*(B = 0) THEN GOTO 350
330 PRINT "MISS"
340 GOTO 290
350 LET S(Z) = S(Z) + 1
360 LET B = B + 1
370 LET X = X + 1
380 LET R(P) = -1
390 GOSUB 650
400 PRINT "COL"
410 FOR I = 2 TO 7
420 PRINT I;"-";
430 IF C(I) = -1 THEN GOTO 470
440 LET C(I) = 10 - I + RND(I)
450 PRINT C(I)
460 GOTO 480
470 PRINT "P"
480 NEXT I
490 PRINT "POTT?"
500 INPUT P
510 IF C(P) = -1 THEN GOTO 270
515 IF Y = 1 AND NOT P = V THEN GOTO 270
520 IF NOT RND(C(P))  1 THEN GOTO 330
530 LET S(Z) = S(Z) + p
540 LET B = B + P
550 IF NOT Y = 1 THEN GOTO 590
560 LET W = W + 1
570 LET C(P) = -1
580 LET V = V + 1
590 IF W = 6 THEN GOTO 650
600 GOTO 120
650 CLS
660 PRINT "SCORE±";S(0);"-";S(1)
670 PRINT "PLAYER±";Z + 1;"±BREAK±";B
680 RETURN
```

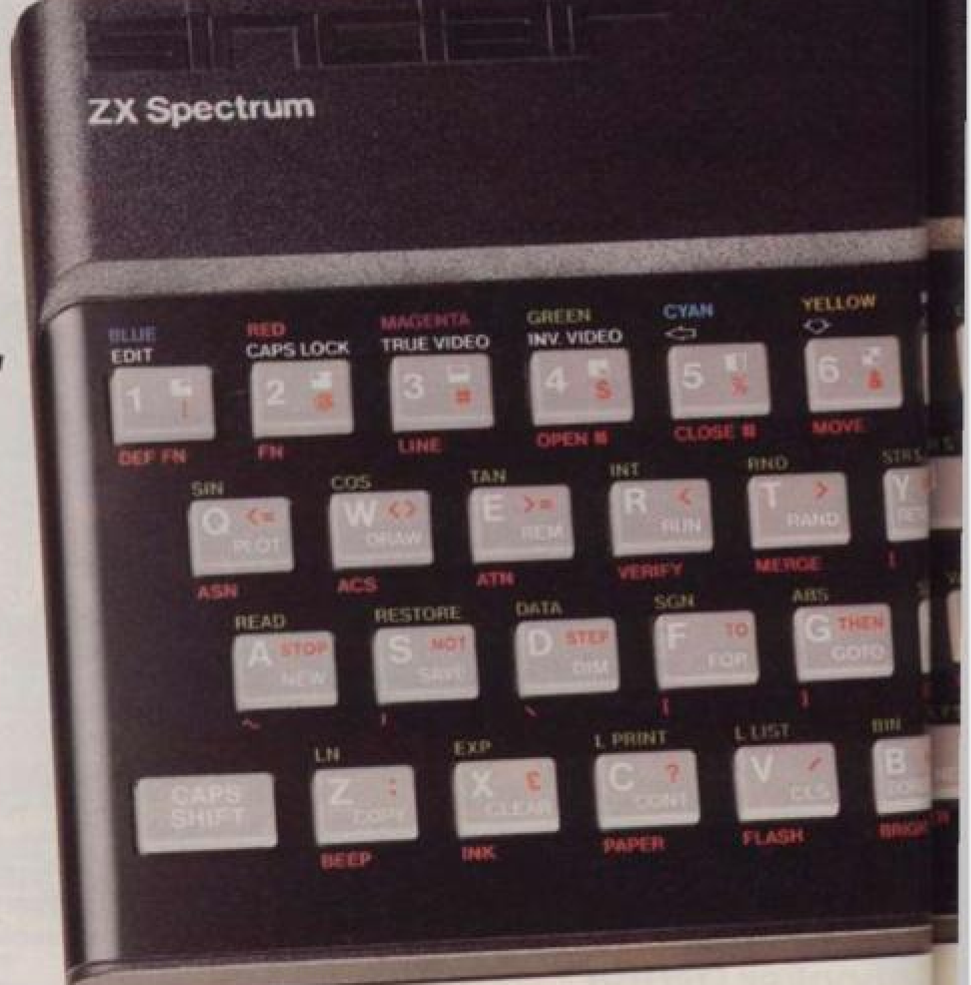
Golf

Eleven-year-old Andrew Haslam, from Rushall, Walsall, has contributed this golf program, which tells you which hole you're on, and how well you're doing. It ends with some 'advice' to make you a better golfer.

Golf Listing

```
20 PRINT, "GOLF"
30 PRINT, "*****"
40 FOR I = I TO 5
50 PRINT
60 NEXT I
70 LET K = 0
80 LET K = K + I
90 PRINT "YOU ARE ON HOLE";K
100 LET Z = RND(5)
110 PRINT "THE BALL IS";
120 IF Z = 1 THEN PRINT "IN THE HOLE"
130 IF Z = 2 THEN PRINT "ON THE GREEN"
140 IF Z = 3 THEN PRINT "IN THE ROUGH"
150 IF Z = 4 THEN PRINT "BEHIND A TREE"
160 IF Z = 5 THEN PRINT "ON TOP OF A TREE"
170 IF Z = 1 THEN LET S = 1
180 IF Z = 2 THEN LET S = 2
190 IF Z = 3 THEN LET S = 3
200 IF Z = 4 THEN LET S = 4
210 IF Z = 5 THEN LET S = 5
220 INPUT A$
230 IF A$ = "S" THEN STOP
240 CLS
250 IF K = 9 THEN GOTO 270
260 GOTO 80
270 PRINT "YOU HAVE FINISHED"
280 LET D = Sx9
290 PRINT "YOU HAVE";D
300 IF D<10 THEN PRINT "SEE YOU AT GLENEAGLES"
310 IF D>20 THEN PRINT "SELL YOUR CLUBS"
320 IF D>9 AND D<21 THEN PRINT "KEEP ON TRYING"
330 PRINT "AGAIN?"
340 INPUT A$
350 IF A$ = "YES" THEN GOTO 10
360 STOP
```

Sinclair ZX Spect



16K or 48K RAM... full-size moving-key keyboard... colour and sound... high-resolution graphics...

From only £125!

First, there was the world-beating Sinclair ZX80. The first personal computer for under £100.

Then, the ZX81. With up to 16K RAM available, and the ZX Printer. Giving more power and more flexibility. Together, they've sold over 500,000 so far, to make Sinclair world leaders in personal computing. And the ZX81 remains the ideal low-cost introduction to computing.

Now there's the ZX Spectrum! With up to 48K of RAM. A full-size moving-key keyboard. Vivid colour and sound. High-resolution graphics. And a low price that's unrivalled.

Professional power–personal computer price!

The ZX Spectrum incorporates all the proven features of the ZX81. But its new 16K BASIC ROM dramatically increases your computing power.

You have access to a range of 8 colours for foreground, background and border, together with a sound generator and high-resolution graphics.

You have the facility to support separate data files.

You have a choice of storage capacities (governed by the amount of RAM). 16K of RAM (which you can uprate later to 48K of RAM) or a massive 48K of RAM.

Yet the price of the Spectrum 16K is an amazing £125! Even the popular 48K version costs only £175!

You may decide to begin with the 16K version. If so, you can still return it later for an upgrade. The cost? Around £60.

Ready to use today, easy to expand tomorrow

Your ZX Spectrum comes with a mains adaptor and all the necessary leads to connect to most cassette recorders and TVs (colour or black and white).

Employing Sinclair BASIC (now used in over 500,000 computers worldwide) the ZX Spectrum comes complete with two manuals which together represent a detailed course in BASIC programming. Whether you're a beginner or a competent programmer, you'll find them both of immense help. Depending on your computer experience, you'll quickly be moving into the colourful world of ZX Spectrum professional-level computing.

There's no need to stop there. The ZX Printer – available now – is fully compatible with the ZX Spectrum. And later this year there will be Microdrives for massive amounts of extra on-line storage, plus an RS232 / network interface board.



Key features of the Sinclair ZX Spectrum

- Full colour – 8 colours each for foreground, background and border, plus flashing and brightness-intensity control.
- Sound – BEEP command with variable pitch and duration.
- Massive RAM – 16K or 48K.
- Full-size moving-key keyboard – all keys at normal typewriter pitch, with repeat facility on each key.
- High-resolution – 256 dots horizontally x 192 vertically, each individually addressable for true high-resolution graphics.
- ASCII character set – with upper- and lower-case characters.
- Teletext-compatible – user software can generate 40 characters per line or other settings.
- High speed LOAD & SAVE – 16K in 100 seconds via cassette, with VERIFY & MERGE for programs and separate data files.
- Sinclair 16K extended BASIC – incorporating unique 'one-touch' keyword entry, syntax check, and report codes.

trum



X Spectrum software on cassettes – available now

The first 21 software cassettes are ow available directly from Sinclair. roduced by ICL and Psion, subjects clude games, education, and business/ ousehold management. Galactic vasion... Flight Simulation... Chess... istory... Inventions... VU-CALC... VU-3D 47 programs in all. There's something r everyone, and they all make full use f the Spectrum's colour, sound and aphics capabilities. You'll receive a etailed catalogue with your Spectrum.

S232/network nterface board

This interface, available later this ear, will enable you to connect your X Spectrum to a whole host of printers, erminals and other computers.

The potential is enormous. And the stonishingly low price of only £20 is ossible only because the operating ystems are already designed into the OM.

sinclair

inclair Research Ltd, Stanhope Road, amberley, Surrey GU15 3PS.
el: Camberley (0276) 685311.

The ZX Printer – available now

Designed exclusively for use with the Sinclair ZX range of computers, the printer offers ZX Spectrum owners the full ASCII character set – including lower-case characters and high-resolution graphics.

A special feature is COPY which prints out exactly what is on the whole TV screen without the need for further instructions. Printing speed is 50 characters per second, with 32 characters per line and 9 lines per vertical inch.

The ZX Printer connects to the rear of your ZX Spectrum. A roll of paper (65ft long and 4in wide) is supplied, along with full instructions. Further supplies of paper are available in packs of five rolls.



The ZX Microdrive – coming soon

The new Microdrives, designed especially for the ZX Spectrum, are set to change the face of personal computing.

Each Microdrive is capable of holding up to 100K bytes using a single interchangeable microfloppy.

The transfer rate is 16K bytes per second, with average access time of 3.5 seconds. And you'll be able to connect up to 8 ZX Microdrives to your ZX Spectrum.

All the BASIC commands required for the Microdrives are included on the Spectrum.

A remarkable breakthrough at a remarkable price. The Microdrives are available later this year, for around £50.



How to order your ZX Spectrum

BY PHONE – Access, Barclaycard or Trustcard holders can call 01-200 0200 for personal attention 24 hours a day, every day. BY FREEPOST – use the no-stamp needed coupon below. You can pay by cheque, postal order, Barclaycard, Access or Trustcard.

EITHER WAY – please allow up to 28 days for delivery. And there's a 14-day money-back option, of course. We want you to be satisfied beyond doubt – and we have no doubt that you will be.

To: Sinclair Research, FREEPOST, Camberley, Surrey, GU15 3BR. **Order**

Qty	Item	Code	Item Price £	Total £
	Sinclair ZX Spectrum – 16K RAM version	100	125.00	
	Sinclair ZX Spectrum – 48K RAM version	101	175.00	
	Sinclair ZX Printer	27	59.95	
	Printer paper (pack of 5 rolls)	16	11.95	
	Postage and packing: orders under £100	28	2.95	
	orders over £100	29	4.95	
			Total £	

Please tick if you require a VAT receipt ☐

*I enclose a cheque/postal order payable to Sinclair Research Ltd for £ ______

*Please charge to my Access/Barclaycard/Trustcard account no.

*Please delete/complete as applicable

Signature

PLEASE PRINT

Name: Mr/Mrs/Miss

Address

ZXC812

FREEPOST – no stamp needed. Prices apply to UK only. Export prices on application.

A little bug

When you're tired of BASIC, you may turn to machine code. Paul Holmes looks at a number of 'tools' to help you with entering and running machine code

Entering machine code using tedious and slow BASIC loaders is tiresome, and you're likely to start looking for something to lend a hand. Here are the latest machine code monitors for the Spectrum.

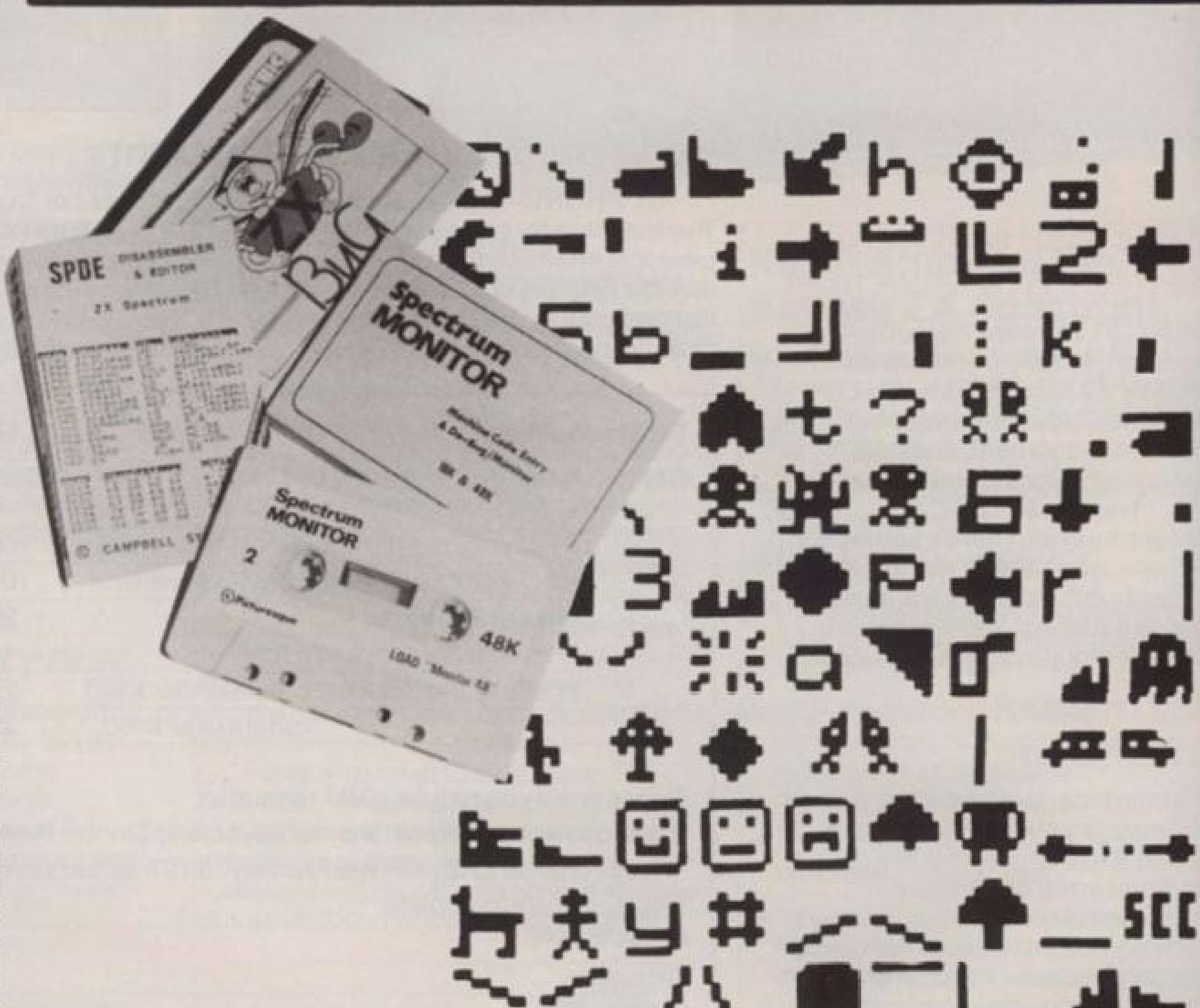
Disassemble Function

One thing which every editor needs is a disassemble option. With this the user can check that all the code he entered was actually the code he wanted to enter, and by disassembling his program he can check it against his original listing. All three of the editors reviewed had this facility, some performed it in a more useful manner by calculating the addresses for relative jumps. Both ZX BUG and SPECTRUM Monitor use 'Z' to access the disassemble funtion. One point about the ZX BUG that was annoying was that to get a printer copy of the disassembly you had to exit (using 'X') then type 'COPY'. This meant that the X command got copied as well, causing a break if a long listing is needed. The SPECTRUM Monitor politely asks 'PRINTER?' in a similar manner to the 'SCROLL?' message used in BASIC.

Instructions

The best manual by far was for the SPECTRUM Monitor by Picturesque. It was pleasantly written and very thorough, the sort of quality one expects when buying a complex package like this. By contrast the instructions for Artic's ZX BUG were squashed onto the back of the cassette inlay. For the experienced user this is just about acceptable but for a newcomer to machine code this could leave them baffled. The commands on all three are accessed by a single keypress, SPDE also has an optional menu to refer to which can be summoned to the screen at any time. The instructions for SPDE are contained in the program and give details of the program's operation and how to use it.

Both ZX BUG and SPECTRUM Monitor, on loading, relocate themselves above RAMTOP and do not depend on a BASIC master program. This gives the user the advantage of being able to write programs in BASIC that use machine code subroutines developed on the editor. With ZX BUG one has to be careful because the machine code is loaded from a dud line (a line with a ridiculously high line number so that it isn't listed) to the top of RAM. That dud line is left behind and so invisibly uses up 4K of your BASIC workspace. To remove it one must first set RAMTOP to 7100 Hex then type NEW. SPDE requires a BASIC master program and so is therefore not quite as versatile.



ZX BUG, or is it SPECTRUM BUG? I'm not sure because my review copy had ZX BUG on one part of the inlay, SPECTRUM BUG on another and ZX BUG 2 on another part. Confusing isn't it? ZX SPECTRUM BUG 2 (GLS!!) had the most functions by far comprising of single and double byte searches, breakpoint system, register display,

String entry, byte modification, copy, byte print (on screen only), character print and many more. One of the commands is a block SAVE command which saves a block of memory in the usual way but can only be reloaded using the complimentary LOAD command (abbreviated to 'L'). No file names are allowed.

One feature which SPECTRUM 2 ZX BUG lacked was a Hex-Decimal/Decimal-Hex conversion function. In my opinion this is a highly useful feature, since at sometime you are likely to want to call your proud piece of code from BASIC using a Decimal number and it is highly frustrating to have to convert it each time for each program or address. Both the other two editors included this feature which I was pleased to see. SPDE seemed to be lacking a comprehensive range of commands and only the actual nitty-gritty seemed to be by machine code subroutines.

Editing and Debugging

Though one would expect the editors for the Spectrum to be an improvement on those for the ZX81 this has not been the case. I used Bug Byte's ZXDB for editing and debugging machine code on my ZX81 and found that editor exceptional, having a single step facility allowing one to monitor everything but the kitchen sink. It was something of a disappointment therefore, when I discovered that none of the three review packages had any such useful features. Being able to trace machine code step by step is immensely useful when bugs arise, so I am waiting with baited breath for Bug Byte's ASPECT Assembler/Disassembler/Editor which will be ready for sale by the time you read this.

There is one improvement I have noticed and that is that the Editors are almost idiot proof, or habit proof. For example, my old editor used to crash when I habitually used the RUB OUT key to delete errors. All the editors had a block move command which is very useful when some code has to be relocated. SPECTRUM Monitor also has an insert command which is very useful when a byte has accidently been missed out or when an instruction must be added. All of the editors have breakpoint systems. A breakpoint system can insert a breakpoint into your program (a point where the program is to be exited), so you can trap errors down to a small part of the program. The breakpoint can be removed and easily relocated elsewhere.

A distinctly annoying point about ZX BUG SPECTRUM 2 was that using addresses (on the 16K Version) 71EO Hex to the top of memory would interfere with ZX BUG and using addresses 4000-6800 Hex or so would interfere with the loader program so that if you typed in some machine code on the earlier mentioned invisible line and then added to, or deleted from the program, your machine code would have moved. This only leaves about 2K of non moving memory for the user even though the invisible line (4K) does absolutely nothing once the program is loaded. None of the Editors had the ability to search for a string more than two characters. SPECTRUM Monitor had no search facilities at all.

It surprised me to learn that Dr Ian Logan uses SPDE since it is remarkably short of editing commands. It can basically do: Hex-Decimal/Decimal-Hex, Block move (in BASIC!), List in Hex and Characters, list as mnemonics and Hex and that's all. It has no Breakpoint system, no fills, no searches, nothing in fact but the bare essentials. It is a pity really because the program itself is nicely presented with a touch of colour here and there and a handy menu to help. Slowly I am getting the impression, after re-reading their letter that Ian Logan is more than mildly acquainted with J.A. Campbell. For instance, he has written a quite complimentary review of their FAST ONE, though I must admit it is worth the cheeky title. Makes one suspicious, doesn't it?

Mixing colours

SPECTRUM Monitor has a blue background which seems fine from both the colour TV owners point of view as well as the Black & White. I say this because our colour TV doesn't like Spectrums and changes white to green and a whole host of colours to pinks, dirty greens etc. This means that I have to use a Black & White (I would probably be ill otherwise), which tends to mix the different colours into an illegible mess.

Since each of the editors has its own merits and demerits I have decided to use a rating table to assist conclusion:

	Content	Presentation	Documentation	Loading	Quirk Free
SPECTRUM MONITOR	****	****	*****	****	****
SPECTRUM BUG	*****	****	*	*****	**
SPDE	**	*****	**	****	****

As you can see, SPECTRUM Monitor has a nice, all round quality and has outstanding documentation which really should be a lesson to other software houses. It's content didn't quite match up to that of SPECTRUM BUG but had sufficient commands to satisfy the average machine code enthusiast. SPECTRUM BUG's documentation was really poor and though SPDE's was better it was still quite sparse. SPECTRUM BUG falls down on the 'Quirk Free' rating because of the floating RAM trouble I explained earlier. All I can say about SPDE is that it is just a pretty face and bad value at £5.95 in comparison with the other two, even if Dr Logan DOES use it. Overall, for ease of use and documentation, the best buy is SPECTRUM Monitor while if you prefer a little more on the commands side then buy SPECTRUM BUG and a magnifying glass. (You'll need it to read the instructions!). SPDE is available from Campbell Systems, 15 Rous Road, Buckhurst Hill, Essex, IG9 6BL at £5.95. Spectrum Monitor is available from Picturesque, 6 Corkscrew Hill, West Wickham, Kent, at £7.50. Spectrum Bug is available from Artic Computing, 396 James Reckitt Ave., Hull, at £7.00

Kayde Graphics Rom

This add-on seems virtually identical to the dk'tronics graphics ROM which has been out for several months now (even down to the 'dk' being one of the graphics available in the ROM). Presumably, as both companies are based in Great Yarmouth they have a deal with each other over the product.

The ROM takes just a few soldering connections to wire up, and if you cannot handle a soldering iron no doubt Kayde would be pleased to arrange for the ROM to be fitted. The ROM sits in memory between 8K and 16K (that is the 8K of memory above the internal ROM and the start of the BASIC program area). This area is used by a few other devices, so check that other peripherals you have are not competing for this slot. Remember, you can only have one peripheral in any given part of the memory map.

The ROM in fact only takes up 4K of this 8K area, and in theory you can fill the other 4K with either 4K of CMOS RAM (6116 chips, easily available for about £4) or a further 4K ROM such as the Kayde or dk'tronics toolkit ROM.

Graphics

The Graphics ROM itself offers some 450 extra graphics which you can use in groups of 64 in place of the standard graphics set. The problem is that you can only be in one character set in the ROM at a time – which means that you cannot have this character from one of the sets of 64 and the other from another set. They have, thankfully, thought out the sets rather well, so that all the Packman figures are in one set, the asteroid ones in another set and so on. A problem occurs when you want lower case letters. The graphics ROM replaces the graphics on your ZX81 and because the graphics on the 'Y', for instance, is just the inverse of that on the 'T', you cannot have both a lower case 't' on the 'T' key *and* a lower case 'y' on the 'Y' key. They get around this by putting the odd lower case letter here and there and leave the user to get used to pressing an odd key for certain letters; this is obviously less than ideal. The only true solution to this problem is to be able to replace all the graphics and their inverses as does the Quicksilver board.

Concluding

A useful addition to your ZX81 then if you have no other add-on in the 8K to 16K region and want true games characters, lower case letters and so forth. Sadly though, unless you purchase the extra CMOS memory you cannot define your own characters, which is often more than half the fun. Perhaps either Kayde or dk'tronics have produced a similar board (for less?) which had only the user definable capability?

The Graphics ROM is available from Kayde, The Conge, Great Yarmouth, Norfolk, NR30 1PJ for £29.95 inc VAT.



*SPECTRUM MONITOR
*SPECTRUM EDITOR/ASSEMBLER
*ZX81 SCREEN KIT 1
*ZX81 ZX-MC *ZX81 REMLOAD

FAST MAIL ORDER-SEND TODAY!

SPECTRUM MONITOR. Machine Code Debug/Disassembler
• Enter, Run, Debug machine code programs. • Compatible with Basic • Breakpoints and Registers Display • Disassembly to screen and/or ZX Printer • 16K and 48K versions on one cassette + 30 page manual. **£7.50**

SPECTRUM EDITOR/ASSEMBLER. A powerful and essential machine code programming aid.
16K & 48K on same cassette with full documentation. Major features include:
EDITOR with Auto Line Numbering; 40 Column screen display, tabulated into fields for easy reading; 5 character Label Names; simple Line Editing and Cursor Control; SAVE/LOAD Text Buffer to cassette; output to ZX PRINTER.
TWO-PASS ASSEMBLER accepts all Z80 mnemonics (plus many unpublished mnemonics); Decimal or Hex numbers; simple arithmetic on operands; Assembler Directives — ORG, END, DEFB, DEFW, DEFS, DEFL, EQU, DEFM.
WE CANNOT FULLY DESCRIBE THIS IMPORTANT UTILITY HERE, AND ASK YOU TO SEND A S.A.E. FOR COMPLETE DETAILS OF THIS AND ALL OUR PROGRAMS. **£8.50**

ZX81 SCREEN KIT 1. More power to your screen in all your Basic programs
BORDERS any size, anywhere on screen. SCROLL in all 4 directions. CLEAR and REVERSE PART OF SCREEN. FLASHING CURSOR anywhere on screen — simulates INPUT. DATA FILES SAVE AND LOAD Basic variables. Double Speed. 880 bytes machine code for INSTANT RESPONSE. Becomes part of Basic Program. **£5.70** 4K to 64K

ZX81 ZX-MC. Machine Code Debug/Monitor
• ENTER, RUN, DEBUG machine code programs • SAVE, LOAD, VERIFY at double speed • BREAKPOINTS and REGISTERS DISPLAY • Self contained — cannot be used with Basic • Cassette plus 36 page manual. **£7.50** 4K to 64K

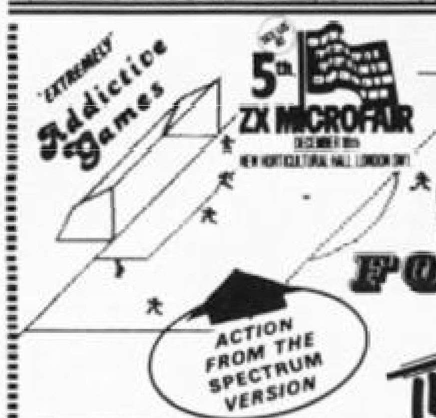
ZX81 REMLOAD. Machine Code Debug/Monitor
• A version of ZX-MC without the Save/Load/Verify facility • Compatible with Basic • CREATE A REM LINE of any length • BREAKPOINTS and REGISTERS DISPLAY • Cassette plus 30 page manual **£6.95** 16K to 64K

SEND S.A.E. NOW FOR DETAILS
6 Corkscrew Hill, West Wickham, Kent BR4 9BB.

Prices include VAT & P&P

PICTURESQUE

SPECTRUM 48K — ZX81 — TRS80 — VIDEO GENIE



NOW! for the SPECTRUM
FOOTBALL MANAGER IN 3D
COLOUR GRAPHICS

FEATURES INCLUDE.......

★ 4 DIVISIONS ★ F.A. CUP ★ PROMOTION & RELEGATION ★ TRANSFER MARKET ★ TEAM SELECTION ★ SAVE GAME FACILITY ★ LEAGUE TABLE ★ AND MUCH MORE!

ALSO AVAILABLE AT BOOKSTORES & COMPUTER SHOPS INCLUDING BUFFER MICRO SHOP & MICROWARE 3D GRAPHICS ONLY INCLUDED IN SPECTRUM VERSION

HARDWARE REQUIRED	
Spectrum 48K RAM	TRS80/ Video Genie
ZX81 16K RAM	LEVEL II 16K RAM

To Order send Cheque/P.O. £7.95 made payable to:—
ADDICTIVE GAMES
at: Dept. Z.X.C. P.O. Box 278
CONNIBURROW,
MILTON KEYNES MK14 7NE
PLEASE STATE COMPUTER

We're tired of telling you how good our game is...... let our customers tell you

I received F.M. about 10 days ago, and have played about 23 hours so far — addictive is the operative word. —
D. BLAIR, FIFE

I think your game is fabulous since starting with Port Vale in the 4th Division, being promoted, winning the F.A. Cup, and being promoted again to the 2nd Division.—
C. DICKENSON, CANTERBURY

I recently bought your F.M. program and was very pleased indeed. I found it very good value for money and played it all day the day it arrived — I own a ZX81 which is now only used for F.M.—
Yours addicted, M. FRAMPTON, CANVEY ISLAND

I felt that I had to write and congratulate you on your program F.M. I found the game very compelling and exciting, really just like the real thing — keep up the good work and standard.—
N. LAWRENCE, FULHAM
ORIGINALS CAN BE SEEN ON REQUEST

MICHAEL ORWIN'S ZX81 CASSETTES

THE BEST SOFTWARE (BY VARIOUS AUTHORS) AT LOW PRICES

QUOTES

"Michael Orwin's £5 Cassette Two is very good value. It contains 10 stolid well designed games which work, offer plenty of variety and choice, and are fun." *Your Computer*, May '82

"Michael Orwin has built a reputation for value-for-money software and his Cassette 4 offers quantity as well as quality." *Sinclair User*, October '82

"If each game was on a separate tape and selling for £5 each I would still recommend them. But all on one for £5 . . . ! This sort of value for money just has not been seen before on any personal computer."
"Without sounding pushy I would like to conclude this review by saying — if you have a ZX81 and like games, then you should buy Michael Orwin's cassette 4." 2 extracts from *ZX Computing*, Oct/Nov '82

CASSETTE 1
(eleven 1k programs)

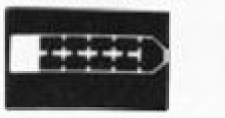
machine code:
React, Invaders, Phantom aliens, Maze of death, Planet lander, Bouncing letters, Bug splat.
Basic:
I Ching, Mastermind, Robots, Basic Hangman PLUS Large screen versions of Invaders and Maze of Death, ready for when you get 16k.
Cassette One costs £3.80

CASSETTE 2
Ten games in Basic for 16k ZX81

Cassette Two contains Reversi, Awari, Laser Bases, Word Mastermind, Rectangles, Crash, Roulette, Pontoon, Penny Shoot and Gun Command.
Cassette Two costs £5

CASSETTE 3
8 programs for 16k ZX81

STARSHIP TROJAN

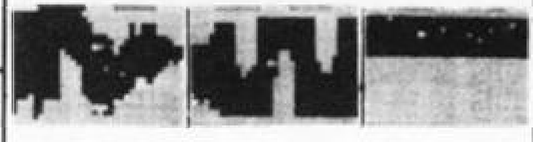


Repair your Starship before disaster strikes. Hazards include asphyxiation, radiation, escaped biological specimens and plunging into a Supernova.

STARTREK This version of the well known space adventure game features variable Klingon mobility, and graphic photon torpedo tracking.
PRINCESS OF KRAAL An adventure game.
BATTLE Strategy game for 1 to 4 players.
KALABRIASZ World's silliest card game, full of pointless complicated rules.
CUBE Rubik Cube simulator, with lots of functions including 'Backstep'.
SECRET MESSAGES This message coding program is very txlp qexi jf.
MARTIAN CRICKET A simple but addictive game (totally unlike English cricket) in machine code. The speed is variable, and its top speed is very fast.
Cassette 3 costs £5

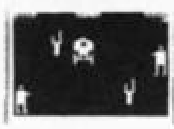
CASSETTE 4
8 games for 16k

ZX-SCRAMBLE (machine code)

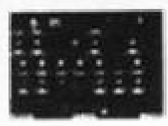


Bomb and shoot your way through the fortified caves.

GUNFIGHT (machine code)



INVADERS (machine code)



GALAXY INVADERS (machine code)
Fleets of swooping and diving alien craft.
SNAKEBITE (machine code)
Eat the snake before it eats you. Variable speed (very fast at top speed).
LIFE (machine code)
A ZX81 version of the well known game.
3D TIC-TAC-TOE (Basic)
Played on a 4x4x4 board, this is a game for the brain. It is very hard to beat the computer at it.
7 of the 8 games are in machine code, because this is much faster than Basic. (Some of these games were previously available from J. Steadman).
Cassette 4 costs £5
FUNGALOIDS (Machine code)

Recorded on quality cassettes, sent by first class post, from:
Michael Orwin, 26 Brownlow Road, Willesden, London NW10 9QL (mail order only please)

Not all that has colour, glitters

In their haste to get Spectrum software on the market, it appears that some companies have simply decided to recycle their ZX81 stuff, adding a few little squeaks from the sound, and an INK or PAPER or three. Phil Garratt takes a close look at some of the material available, and notes that among the dross, gold still shines.

"Space Intruders" — Quicksilva

The race to put new home computers on the market is matched only by the software companies race to be the first to market versions of the "standards" — Invaders, Asteroids, Defender etc. for each new machine. There must be a strong temptation to put speed before quality, but fortunately Quicksilva have produced two arcade games for the Spectrum as good as anything they produced for the ZX81.

"Space Intruders" is the best version of Invaders that I have yet seen on any home computer. There are three types of invaders, plus a wobbly "mutant" which you can try to zap before it turns into an ordinary invader. Each hit is marked by a little starburst, including ones on the command ships which cross the screen in both directions.

The game gets progressively harder by speeding up and by allowing the invaders to drop a greater density of bombs at you. The bombs fall at two speeds, and you have to be on your toes to avoid the quick ones. There are several features to help you win a high score — the fire button "auto-repeats", an extra base is earned at 50,000 points, and if your base does get hit, all the invaders' bombs are cleared from the screen to give your new base a fighting chance. In addition the game can be "frozen" on the screen by pressing "h" and then continued with "s". We didn't have this kind of luxury in my day — you needed endurance to win a high score then! Unfortunately the program does have one bug — every hit scores 660 points whatever kind of invader is knocked out. I found this rather spoilt the game for me as it made me change my tactics since there was no point in chasing mutant invaders.

Excellent use is made of the limited Spectrum sound effects — by using warbling rather than steady tones there is no slowing of the action. On-screen scoring and highest score, plus a high-score table, full use of colour, and "self-play" demonstration mode make this a really excellent, full-featured Invaders.

"Space Intruders" is £5.95 from Quicksilva, 92 Northam Road, Southampton SO2 0PB.



"Meteor Storm" — Quicksilva

The second of Quicksilva's arcade games for the 16K Spectrum is "Meteor Storm", their version of Asteroids. If you're one of those who belittle the "beep" on the Spectrum, this program will make you sit up and listen. Never mind the Japanese fifth generation computers, the Spectrum does it already — it talks to you! Admittedly it helps if you amplify the sound and you know what it's going to say ("meteor alert!"), but it is intelligible and a remarkable feat of programming, even if it does sound like a Dalek with a sore throat.

There are three sizes of meteors, the smaller ones worth more points, and if you clear one lot even more appear. Your controls are "6" for anti-clockwise and "7" for clockwise, "8" is hyperspace (a random jump to somewhere else on the screen, which can be used repeatedly). "9" is thrust and "0" is fire, which has no auto-repeat, so fortunately this game also has the "hold" and "start" feature on "Intruders", allowing you to rest your finger.

Bonus points can be won by knocking out the enemy saucers which appear from time to time and start shooting at you. There is a slow moving large one and a quicker small one, both of which look remarkably like raspberry pies! Every 10,000 points scored results in a noise which sounds like someone treading on a piglet's trotter, and more importantly you get a bonus ship.

On-screen scoring, high score table, good sound effects and demonstration mode complete another winner from Quicksilva. But watch out for low flying raspberry pies!

"Meteor Storm" is £5.95 from Quicksilva, 92 Northam Road, Southampton SO2 0PB

"Great Britain Limited" — S W Hessel

Is there anybody who doesn't think they could do a better job of running the country than the government? Well, all you need is a 48K Spectrum and "Great Britain Ltd" and you can find out.

You are Prime Minister and Chancellor rolled into one, and you can choose whether to represent Labour, Liberal, Conservative or SDP. Your task is to juggle the economy with the aim of staying in power for as long as possible.

At the start of the game the state of the nation is displayed, with information including the inflation rate, unemployment and your popularity. The Country Profile tells you the number of pensioners, children and companies, plus a breakdown of the revenue from all the various taxes. The Shopping Basket gives you the current prices of household items and the Retail Price Index. You then get your chance to meddle!

The tax and duty rates are displayed and can be amended, as can the welfare benefits which follow. Failure to maintain benefits can lead to riots and worse, so watch out. You are also given the chance to carry out social reforms such as expanding the Health Service or building new schools. As these boosted popularity, I found it best to time them as close to the election as possible (a trick I learnt from the real politicians!). If you manage to last five years, a General Election is held, which usually led to me being booted out, often in favour of some rather odd coalitions (Labour-Conservative!?!).

Inflation Jumping

"Great Britain Ltd" is a game and not an attempt at realistic economic forecasting. Unlike real life, the effect of decisions is immediate and often drastic, such as inflation jumping from 10% to 120% in one year. But enough information is presented for a player to make very careful decisions based on the data, if he or she wishes to take it seriously. There are many ways in which this game could have been made more realistic and sophisticated, but the program would become much slower to run. I think the author has struck a good balance, and produced a difficult and challenging game.

"Great Britain Limited" is £5.95 from S W Hessel, 15 Lytham Court, Cardwell Crescent, Sunninghill, Berks.

```
*******RT.Hon.Fred Bloggs*******
TERM:1  **GREAT BRITAIN**   YEAR:2
INFLTN:18 Percent*EXCH.RTE:$1.73
UNEMPT:1.2Millions*POP.RTNG:14
REFRMS:0  ACC.BALANCE:£M-14543

         COUNTRY PROFILE
      55M People (including)
 (10M Children + 12M Pensioners)
        949000 Companies
        INCOME FROM TAXES
   VAT Income             £   15140M
   P.A.Y.E. Income        £   66208M
   Company Tax Income     £   58858M
   Excise Duties          £    9048M
   TOTAL INCOME           £  149254M
   less EXPENDITURE       £  157241M
                             ==========
   BALANCE                £    -7987M
================================
```

```
*******RT.Hon.Fred Bloggs*******
TERM:1  **GREAT BRITAIN**   YEAR:2
INFLTN:18 Percent*EXCH.RTE:$1.73
UNEMPT:1.2Millions*POP.RTNG:14
REFRMS:0  ACC.BALANCE:£M-14543

     BUDGET DAY i.Taxes

     Category..........Rate:

     a.Income Tax        30%
     b.Corpn. Tax        50%
     c.VAT               15%

     d.Tobacco Tax     £0.6
     e.Alcohol Tax     £4
     f.Petrol Tax      £0.8
     g.NO MORE CHANGES
================================
 Enter PREFIX of tax to change
```

```
*******RT.Hon.Fred Bloggs*******
TERM:1  **GREAT BRITAIN**   YEAR:2
INFLTN:18 Percent*EXCH.RTE:$1.73
UNEMPT:1.2Millions*POP.RTNG:14
REFRMS:0  ACC.BALANCE:£M-14543

     BUDGET DAY ii.Benefits

     Category...........Rate:

 a.Child Allowance       £5.5
 b.Old Age Pension       £30
 c.Unemployment Benefit  £28
 d.NO MORE CHANGES
================================
```

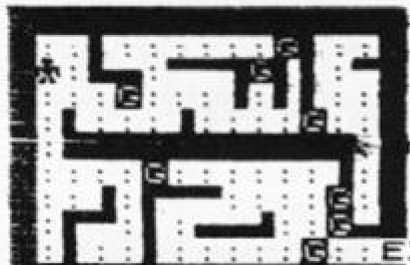
"Venture" — G & J Bobker

ZX-Guaranteed's first foray into the Spectrum market is "Venture", a program made up from seven separate games rolled into one. You start with £1,000 and at each stage you may either win some money or lose half of what you have. The first game is "Duckrace", in which three ducks move randomly across the screen, and you have to back one to arrive first. About as sophisticated as a boiled ducks egg, and the second part, "Las Vegas" is much the same. You are given ten goes on a one-armed bandit — pressing "s" stops the reels, but there is no hold feature. I can hardly think of a better application for user-defined graphics than a fruit machine, but unfortunately the program uses flickering numbers instead.

The third part is "Treasure Hunt" in which you have to guess the position of the treasure on an 8 x 8 grid. The game is made very easy by the clues given (e.g. "down and right"), so it's not much of a challenge. The next game is "Mastercode" — again numbers-based and a completely routine Mastermind.

Racing cars

The fifth game, "Track", is an obvious conversion from the ZX81 — "5" and "8" steer a racing car on an upward-scrolling road. Leopards don't change their spots, and a flickery ZX81 game remains just that even if it is put onto the Spectrum. In the sixth part, "Bomber", you have to bomb a ZX81 before your plane crashes into it. This strikes me as being a rather ungrateful way to treat the machine on which all these programs were originally written!



```
                        CASH=£2475
                        GOLD=£0000

                 EXIT   YOU HAVE

                        1.CHEESE
                        2.COIN
                        5.TORCH

HERE IS A AEROSOL PAINT

T=TAKE.  L=LEAVE.


 'VENTURE' - ZX-GUARANTEED
```

"CASSETTE 1" — Silicon Software

"Elephants Graveyard" is an adventure-type game in which if you can survive five weeks in the jungle you will find yourself at the legendary graveyard. You start with 100,000 Kes (the local currency) with which you can purchase the assistance of natives, as well as supplies of food, guns and tents. Running out of supplies or money to pay the natives leads to an early demise, so a careful balance has to be found at the start. As you progress through the jungle, unpleasant things happen at random — attacks by lions and snakes, quicksand, storms, losing the trail and such like.

The game looks like another ZX81 conversion — it has no user defined graphics and very little sound or colour. There are some minor bugs which suggest *it hasn't been thoroughly* tested.

Also on the cassette is "Sales", a marketing simulation game for any number of players,

in which you are in charge of an ice cream stall. You start with a working capital of £2 and have to decide how much to spend on advertising and ice cream stock based on the day's wheather forecast. I usually ran into a severe liquidity crisis — all my unsold ice cream melted!

Apart from the opening titles, no colour, graphics or sound are used. This program would run as happily on an old-fashioned teletype as on a full colour bells and whistles home computer. Both programs will run on a 16K or 48K Spectrum.

"Cassette 1" is £2.95 from Silicon Software, 24 Short Lane, Stanwell, Middlesex TW19 7BQ

```
Day: 2  Stall: 1

Assets: £1.82

Advertising signs cost 10p each.

Ice pops cost 4p each.

Press "W" for a weather report.

The weather forecast:-

          COLD AND WET
```

Vampires and Gargoyles

"Quest" is a non-graphics adventure game with random locations and monsters. It is played in real-time, so if you stop to think you will be pounced on by a gargoyle or vampire. The delay loops for the command entry have been poorly programmed, so the timing of key presses is critical. I found nearly all of my commands being ignored until I inserted an extra pause statement. This relentless and repetitive game left me rather frustrated.

In "Reverse", nine random digits have to be juggled into order. It appears to have been included to make up the numbers, and certainly isn't of commercial quality.

"Startrek" is a version of the classic galactic war game. The short range scan gives a graphic display, but for some reason user-defined graphics have not been used for the Enterprise, Klingons or starbases. Phasers still shoot at Klingons after they have been destroyed with Proton (sic!) torpedoes, so you can end up with minus 1 Klingons on the long-range scan. Perhaps they are anti-matter Klingons! A very limited version of the game.

All four programs make little or no use of the extra graphics and sound facilities on the Spectrum, and show signs of being hastily put together. Serious Software? — they must be joking.

"Spectrum Games" is £6.95 from Serious Software, 7 Woodside Road, Bickley, Bromley, Kent BR1 2ES.

'Startrek' – Serious Software

```
Location

left:    stairs up
forward: wall
right    corridor

enter command
a giant rat has appeared!
it is 3 feet away from you
make a move before it finds you
```

'Quest' – Serious Software

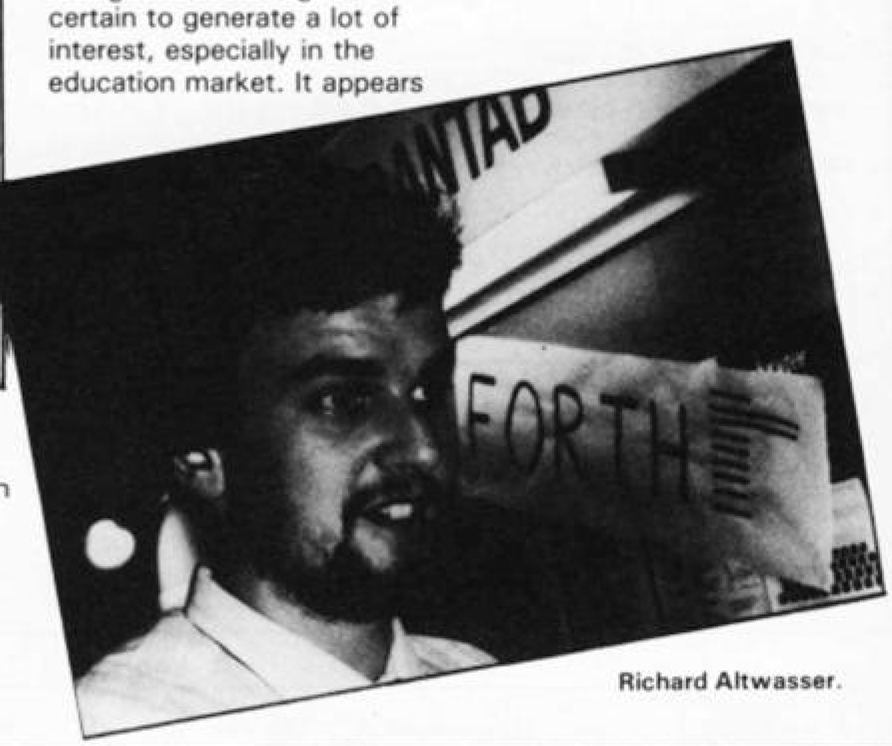
Rival Systems Grow



It may well be that the reign of the Sinclair computers is over — as more and more low cost systems come on the market.

Richard Altwasser and Steven Vickers, the two principle people behind the Spectrum, have released their Jupiter Ace, a computer which uses FORTH, in contrast to most personal computers which have BASIC as their main language.

The Jupiter Ace looks suspiciously like a ZX81 with a Spectrum keyboard. It is black and white, has ZX81-like graphics on its keys (which tend to follow the standard Sinclair layout, in terms of the position of things like the question mark, colon and greater than signs), and is certain to generate a lot of interest, especially in the education market. It appears that Jupiter Cantab, as the Vickers/Altwasser company is called, seems set to repeat another Sinclair characteristic — long delivery times, but this may only be teething troubles.



Richard Altwasser.

The Japanese invasion

A host of micros for under £200 will be launched at the British market in the next three months, from manufacturers like Casio and Sharp, and each of these could significantly erode Sinclair's impressive lead.

Sinclair, however, is not sitting idle.

Under the energetic hand of Bob Denton, the company Prism Microproducts, has been pushing the £49.95 black and white ZX81 into every outlet it can find. Dealer ads like the one illustrated are pushing the Sinclair gospel hard.

After a long period with exclusive retail distribution, Smiths now share the shop floor on ZX81's with a growing band of outlets. The major ones to have the machine include branches of John Menzies and northern chain Wigfalls.



Oye, Oric

Tangerine have just launched their new £99 computer, Oric 1, which will pose a much greater threat to the Spectrum than does the Ace, because Oric's specifications compare very favourably with the Spectrum.

A new company, called Oric Products (funded by British Car Auctions!), has been set up to push Oric 1 through mail order, and through the retail trade. The company claims, naturally enough I guess, that their computer is better than the Spectrum but think that once Oric is seen as getting a market lead, Sinclair will respond by cutting the price of Spectrum. The 16K Oric is £99 and this will be available by mail only. A 48K version for £169 will be available from retail outlets. A Prestel modem is planned which should sell for around £60.

Oric 1 has a Centronics interface for standard printers, standard Microsoft BASIC, a 16-colour display, and what the manufacturers claim is 'hi-fi sound'.

Dragon bites hard

Although the Dragon 32 has been criticised in some quarters for being a little overpriced for its capabilities, and for being built around a relatively 'old' ROM (much the same as the one in the Tandy Colour Computer, which has been out for a couple of years), ZX Computing editor Tim Hartnell says the Dragon has made quite dramatic inroads into the popular computer field.

Part of this inroad is due to its wide availability, and the way it is being pushed at the retail level.

At a microcomputer show in Croydon late in September, the quantity of Dragon support material — most notably from Premier Publications, a major software firm which has started a Dragon club — suggests that more and more people will be attracted to the machine, simply because the software backup is there.

When he was recently interviewed on BBC radio, Tim Hartnell was somewhat surprised to find that of the four people in the studio — none of whom knew too much about computers — two were thinking of buying a computer, and the machine they were planning to get was the Dragon.

Whither Binatone?

The promised £49.95 Binatone colour computer seems less of a figment of some PR writer's imagination, and more of a reality, following the disclosure by Binatone managing director, Partap Lalvani, that their computer will be built by the Hong Kong company, Electrophonic Corporation, who made a name for themselves with a low-cost product which closely resembled the Sony Walkman personal cassette system.

The Binatone computer looks like it will have a Spectrum-like keyboard (although somewhat larger), but will offer exactly the same facilities as the Dragon, which costs four times as much.

It is understood that the Binatone machine will be built around the same ROM as the Dragon.

School support

To increase its direct support to schools Sinclair has appointed E J Arnold & Sons, to act alongside its existing distributor, Griffin & George. It is also in advanced negotiation with other leading educational distributors.

For the general promotion of microcomputing in schools Sinclair is producing, in association with the Educational Foundation for Visual Aids, a special videocassette which will be made widely available to local education authorities and institutions.

Finally to coordinate its increased activity in the education sector Sinclair has appointed David Park, who was with Sinclair between 1969-79, as educational marketing manager, and John Wright as educational consultant. A former primary school headmaster, John has been widely active in the promotion of school's microcomputing.

The Department of Industry's 'Micros in Primaries' scheme was announced by the Prime Minister on July 16. It is estimated to be worth up to £9 million, and will operate from October 1 to the end of 1984. 27,000 primary schools in the maintained and independent sectors are eligible for 'pound-for-pound support' and should apply through their local education authorities.

The scheme offers packages based on three microcomputers; Sinclair ZX Spectrum (48K memory version), BBC Model B and Research Machine's 480Z. The Spectrum package comes complete with 'Horizons' welcome tape to introduce the computer, RS232 interface board, cassette recorder, Microvitec colour monitor or black and white TV, custom-built tray to house the equipment, manuals and leads.



Taking control

The new ABACUS CONTROLLER for the ZX Spectrum has the SAVE and LOAD modes of the orginal model, eliminating the tedious swapping of plugs when saving and loading programs on cassette.

The added feature is the AMP mode which switches in the built-in amplifier and loudspeaker to boost the weak Spectrum sound output. The volume can be adjusted by a control at the rear of the unit. The unit is powered off the Sinclair power pack. All connections are plug-in, therefore no soldering is necessary. The plugs that go from the controller to the cassette recorder are standard 3.5mm jack plugs. If the customer has a DIN or any other connection on his cassette recorder, we will happily supply the appropriate plugs. The controller has already met with great demand and is available direct from Abacus Electronics or over the counter at the Buffer Microshop in London

The new controller is priced at £14.95. The original controller for the ZX81 is still available priced at £12.00 inclusive.

The ABACUS CONTROLLER is made by Abacus Electronics, 186 St Helens Ave., Swansea, West Glamorgan, SA1 4NE.

Big growth predicted



Clive Sinclair

Whether it is Sinclair that gets the lion's share of the market, or some other manufacturer, there seems no doubt that the micro market is going to keep growing at its present explosive rate.

John Golding, UK manager of Hewlett-Packard's personal computer group, predicts that personal computers will grow at a compound rate of 130% a year to 1987.

He says that by then, a third of the £7,000 million spent that year worldwide on personal computers, will go on portable ones.

Understandably, his comments came at the same time as he unveiled a new computer from his own company, the £693 HP75C portable machine, which looks like a slightly smaller Spectrum, modelled after the handheld Sharp PC-1211. It has a single line display, magnetic card reader, and 16K on board. Peripherals like a full-size monitor, printer and digital cassette unit will add another £250 to the price.



Going to school

Of course, Sinclair has also got the advantage of a belated government backing for his computers.

Some 27,000 primary schools are now eligible to get government assistance in buying a Spectrum, and Sinclair themselves have added extra frosting to the government cake.

Sinclair's 'frosting' scheme extends the Department of Industry's new £9 million 'Micros in Primaries' project, and is part of Sinclair's continuing commitment to education.

Schools ordering Sinclair's ZX Spectrum personal computer under the government scheme will also receive a free Sinclair ZX Printer, a free copy of the LOGO computer language, and ten special discount vouchers, valid until the end of 1984.

One voucher may be returned with each further order for a 48K Spectrum. Schools may opt to use the voucher either to obtain a £45 discount from the Spectrum's normal price (£175), or to pay the full price and receive with their order a free ZX Printer, worth £59.95, both prices inc VAT. In total Sinclair's contribution is worth up to £15 million.

Speaking at a London press conference, Clive Sinclair explained: "We were delighted to participate in the Department's scheme, and fully share its declared objective, 'to give all young pupils experience with technology'.

"Believing that one micro per school is inadequate, our scheme seeks to make computers available at low-cost to provide sufficient numbers for all pupils to have practical 'hands-on' experience".

The new offer is part of a comprehensive Sinclair package, covering equipment, software, personnel and technical support, which will further the development of microcomputing in primary schools.

Provision of LOGO and, at a later date, PROLOG will make available two of the most popular computing languages for junior education. At the same time, recognising schools' urgent requirement for software packages, Sinclair is cooperating with educational specialists to expand its own range alongside the government scheme's 150-program library.

Press and tell

At a recent press conference, Sinclair Research's Nigel Searle intimated that a super-low-cost Prestel adaptor for the Spectrum would be available in due course. He didn't say how much it would be, but it sounded as if he was hinting it could be as low as £20.00.

The press conference was notable not for the announcement, but more for the impression it left in journalist's minds that the purpose of it was more to discourage other people who were working on Spectrum-Prestel interfaces, than to announce anything.

The announcement came at a time when interest in the interaction between micro users and Prestel is increasing.

Bob Denton (already mentioned in connection with Prism, the company which is flogging ZX81's to the retail trade), has also teamed up with Richard Heese to form a company to set up a giant micro database on Prestel. They've offered user clubs free pages in return for providing software for the pages.

At the same time, British Telecom have announced that they've decided to give residential users of Prestel 'free' access to the system from 6pm to 8am Monday to Friday, from 1pm on Saturday, and all day Sunday.

There will be no time connect charges if the system is accessed in these times. The whole lot will be covered by a quarterly subscription charge of £5.00.

The non-business use of Prestel has been most disappointing to Telecom (some sources suggest there are less than 3,000 personal subscribers in the whole country) and this new price scheme is intended to try and bolster the systems flagging fortunes.

If you get hold of a Prestel system, start at Aladdin's Cave (page 700) and you'll get access to a wide range of ZX81 and Spectrum information.

Crowded scenes at the New Horticultural Hall.



New Microfair venue a success

The ZX Microfairs, which have suffered from overpopulation problems in the past, have found a venue sufficiently big to take the crowd — the New Horticultural Hall, Victoria, as this photograph of a recent, and most successful fair, shows.

The next Microfair will be held at the same venue, a week before Christmas.

Making the most of tape

To help owners of small computer systems make the best use of tape as a medium for outside storage, Mike Salem, head of the ZX business software company Hilderbay Ltd., has written and published a book called 'The Microcomputer User's book of Tape Recording'.

This 60-page book, which costs £2.90, covers such subjects as 'choosing a tape recordings' and one subject of tapes', 'making reliable recording' and one subject sure to interest ZX81 owners, 'loading "difficult" tapes'.

Did I say that?

While looking through some old magazines, a fascinating interview with Clive Sinclair was found.

In 1977, Uncle C, looking dapper in tinted glasses, a wide floral tie, and with four wisps of hair plastered vertically down his forehead is quoted as follows:

"We're not basically in the business of the hobbyist these days . . . although we tend to make products which are technologically interesting . . ."

Speaking of the 'Black Watch Disaster', he said: "What went wrong was one of those things that happens now and again, and it was tough luck . . ."

His comments on another product which did not meet full expectations, the PWM amplifier: "We didn't test it thoroughly enough before we launched it, but we soon put that right."

The article, in a special issue of Electronics Today called "Electronics Tomorrow" continues: "We asked Clive if he had got over the bad name he once had. In the hi-fi days, it was commonly rumoured that he had a 20% failure rate on some of his products. Mr Sinclair said: 'True. The quality control wasn't as good as it should have been, there's no doubt about it . . . Our quality is superb now . . ."

And what about home computers? Oh Clive, did you really mean it when you said: "I think there's not much use for the computer in the home . . . In 1984 (and remember, he was talking in 1977) we'll be more into computers than we are now. We see ourselves as in on the beginnings of computers with the programmable calculator. I'm sure that we will have gone a lot further by then."

Machine code test tool

'Machine Code Test Tool' is the title of a handsomely-presented cassette and booklet for the 16K ZX81 which has been prepared in order to give practical examples and 'hands-on' experience of using machine code programs, as well as to examing the ZX81 ROM, and look at how BASIC programs are stored in RAM. The 32-page booklet, written, (as was the program) by Francis Ainely of Oxford, seems a careful, accurate introduction to Z80-based machine code. 'Machine Code Test Tool' is £9.95, and is available from F Ainely, 76 Linkside Ave., Oxford Ox2 8JB.

ROM-based books

Researchers at the University of Colorado are working on tiny 'books' which are held in ROM form, and then made available, a page at a time to readers.

Normally, it takes a lot of memory to hold a book, around £150 worth of the new, £1 64K ROM chips.

However, the boffins at the university have decided that the best way round the problem is to have a ROM which holds the 2,000 to 4,000 words which comprise more than 90% of written English, and a second 'book' ROM which stores the other 10% of words, and 'calls' the ROM-stored ones when needed.

With this technique, they hope to be able to bring the price of ROM-books down to a point where they would be commercially attractive. Now if Uncle C can only get his flat-screen telly act together, we'll soon be 'reading' ROMs on the train.

Getting jobs

Uncle C's company is backing the Daily Star's 'Get Going' campaign to find worthwhile jobs for teenagers.

To help and encourage youngsters facing the ever-growing dole queues, the Daily Star is giving away £10,000 to get the best ideas for creating jobs off the ground, and Sinclair Research has pledged support adding £1,000 to the prize fund.

Prime Minister Mrs Margaret Thatcher has already backed the scheme

Spectrum book explosion

Every publisher in the country seems to have discovered the Spectrum, and a flood of books — which is by no means at its height — is filling specialist and general bookshops.

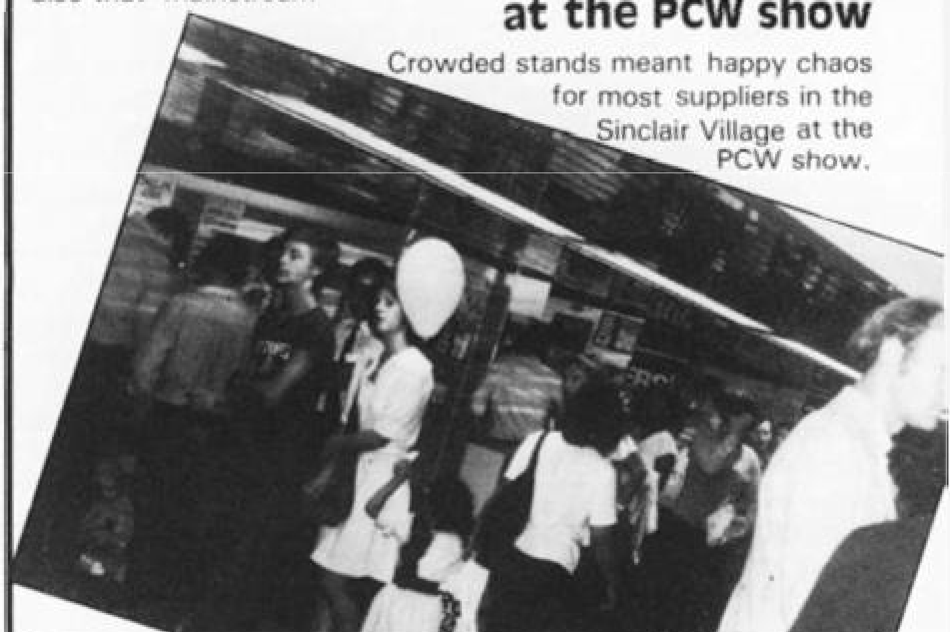
The books include Shiva's 'Computer Puzzles: For Spectrum and ZX81' and 'Easy Programming for the ZX Spectrum', both by Ian Stewart and Robert Jones; Granada's 'The ZX Spectrum and how to get the most from it'; Interface's 'Programming Your ZX Spectrum', by Tim Hartnell and Dilwyn Jones, and 'The Spectrum Software Library' by David Harwood and Richard Altwasser's 'The Cambridge Colour Collection'.

Of particular interest is not only that the general standard of presentation is improving (and our reviewers look at the quality of the contents in the book review section of this issue of ZX Computing), but also that 'mainstream' publishers are recognising the potential value of having Spectrum titles in their list.

Fontana Paperbacks will shortly publish a book by Harrow student Robert Speel, on better programming on the Spectrum and ZX81, and Penguin Books have a title in the pipeline.

Sinclair's own publishing company, Sinclair Browne, has a series of books under the general title 'Sinclair Computer Guides' (which include such esoterica as 'Computers for Farmers'). The Spectrum title, written by Tim Hartnell with a foreword from Uncle Clive himself, is 'The ZX Spectrum Explored', and as well as the expected contents ('making the most of sound', 'how to write games programs' and the like) includes a chapter on 3-D graphics, and one which is intended to act as an introduction to machine code.

Sinclair Village proves a hit at the PCW show





Crowded stands meant happy chaos for most suppliers in the Sinclair Village at the PCW show.

User's clubs

The number of user clubs for ZX enthusiasts continue to grow. The National ZX User's Club (44-46 Earls Court Road, London, W8 6EJ) acts as an umbrella club, maintaining a comprehensive list of every local club it can locate. The club is giving away a book of 20 programs (10 for the ZX81 and 10 for the Spectrum) to each new member. One pound will get you a sample copy of the club magazine, plus details of the services it offers members.

Local clubs include the following:

The North London Hobby Computer Club has ZX80/81 users group meeting each Monday night from 6-9pm. It is held at the North London Polytechnic, Holloway Road, London N7 (diagonally opposite Holloway Road tube station). Tel:01-607 2789.

Other local groups:

- Roger Pyatt, 23 Arundel Drive, Orpington, Kent (66) 20281
- Austin Knott, 269 Telegraph Road, Deal CT14 9EJ.
- Christopher Moeller, Gross Kurfurstenstasse 41a, 4800 Bielefeld 1, Germany.
- Danmarks Nationale ZX80 og ZX81 Club, Skovmosvej 6, 4200 Slagelse Dk. Denmark.
- Steve Brumby, 38 Eastfield Road, Messingham, Scunthorpe, South Humberside.
- David Blagden, PO Box 159, Kingston-upon-Thames, Surrey KT2 5YQ.
- Anthony Quinn, Heckenrosenweg 6, 3170 Gifhorn, W. Germany.
- Conrad Roe, 25 Cherry Tree Avenue, Walsall WS5 4LH.
- Ian Watt, 107 Greenwood Road, Clarkeston, Glasgow.
- J.Palmer, 56 Meadowfield Drive, Edinburgh. (031-661 3181).
- Leeds Microcomputer Users Group. Meets fortnightly on Thursday evening in Leeds. New members welcome. Contact: Paul O'Higgins, 20 Brudenell Mt., Leeds 6. Tel:(0532) 742347 after 6pm.
- Brunel Computer Club: meets alternate Wednesdays, 19.00-22.00 hrs at St Werburgh's Community Centre. Contact: Mr R. Sampson, 4 The Coots, Stockwood.
- Worle Computer Club: meets alternate Mondays, 19.00-22.30 at Woodsprings Inn Function Rooms. Contact: S. Rabone, 18 Castle Road, Worle, Weston-Super-Mare, Avon. Tel:0934 513068.
- P. Compton, 29 North Marine Road, Scarborough, North Yorks. YO12 7EY.
- Alan Gunnell, 66 Nursery Road, Hookend, Nr Brentwood, Essex
- Jonathan Meyer, Vanspaen Straat 22, 6524 H.N. Nymegen, Holland
- Raymond Betx, Chemin du Moulin 38, 1328 Ohain, Belguim.
- ZX Microcomputer Users Group: Paul King, 25 Fir Tree Way, Hassocks, West Sussex Tel: Hassocks 4530.
- The Inverclyde ZX User's Club meets fortnightly on Mondays at Greenock in the halls of the Greenock Society for the Deaf, Kelly Street. Robert Watt (Tel: 30067 evenings) has full information.
- The Gwent Amateur Computer Club: Ian Hazell (Secretary), 50 Ringwood Hill, Newport, Gwent NPT 9EB.
- ZX81 Datorklubb, c/o Kenneth Nilsson, Drottninggarden 244, S-261 46 Landskrona, Sweden.
- L. Henson, 2 Lark Down, Trowbridge, Wilts. BA147JX. (Tel: Trow 67477).
- The Aylesbury ZX Computer Club: contact Ken Knight on Aylesbury (0296) 5181 or write to him at 22 Mount Street, Aylesbury, Bucks. HP20 2SE.
- Harlow Micro User's Club: meetings are at Kingsmoor House, Parringdon Road, Harlow, from 7pm to 9.30pm every Monday night.
- The 81 Club: Mike Hayes, 54 Oakley Place, Grangetown, Cardiff. Tel: Cardiff 371732.
- New Brighton Computer Club, Merseyside: 051-639 6712.
- Keighley Computer Club: Colin Price, Red Holt, Ingron, Keighley. Tel:0535 603133.
- Mid-Cheshire Computer Club: Dave and Liz Clare, 222 Townfields Road, Winsford, Cheshire.
- South Trafford Computer Club: Contact Dave Edwards on Manchester (061) 969 3317.
- Royston H. Wallis, 22 Mallard Crescent, Pagham, Bognor Regis, West Sussex. PO21 4UU. Tel: Pagham 66795.

Harrogate

The Harrogate ZX Users Club meet at the P.H.A.B. Club in Harrogate. Details of the club can be got from the Harrogate branch of W.H. Smiths. You could also contact Mr S. Atkinson at Flat 3, 3 Heywood Road, Harrogate.

Belgium

P. Glenisson wants to make his club known to as many Belgium ZX users as possible. The address to contact is Priester de L'Epeestraat 14, 1200 Brussels, Belgium. Oh yes, the name of the club is the Belgian-Dutch Microcomputer Association.

Orpington

The Orpington Computer Club meets every week on Fridays and claims to be open 365 (count 'em) days of the year. Associate membership of the club is available at £8.00 per annum.

For further details get in touch with Mr J.P. Gibbon, 14, Avalon Road, Orpington, Kent BR6 9AX.

Footy pools

Some computer games can go cold after a while, so here are two interesting non-games for the ZX81. The Z4 Football Pools Forecast and the H5 Horse Racing Forecast are two new programs for the ZX81 16K machine that promise to offer not only endless entertainment but also a chance to perhaps win the pools or bring in a big win on the horses.

These large and well-documented packages are data or form processing and analysis systems that are simple and easy to use. You simply enter the performance of the football teams, for example, and your ZX81 will analyse all your input, including an interpretation based on your own hunches or inside information, such as 'star player has broken leg', and it will output its's analysis of wins, draws, homes, aways, and so on. All you have to do, the suppliers tell us, is to fill in the coupon and decide what to do with all that money!

The H5 Horse Racing Forecast Program works in a similar way. Once you have put in the merits, form, performance of any number of horses in any particular race, the options for you to back are displayed. Another tip from Holly Products, the supplier — 'Don't tell your bookmaker, he might not pay out with all that memory giving you an unfair advantage'.

The man behind these complex statistical analysis programs is Professor George, an expert and lecturer on Statistics and how to make them work to provide useful and interesting results like 'What might win the 2.30 on Saturday'. That's useful!

The Z4 Football Pools Forecast and the H5 Horse Racing Forecast Programs are available, fully documented, from: Holly Products, Blackthorn House, Dukes Lane, Gerrards Cross, Bucks.

Expandable RAM

A new RAM pack, called the ZX-Panda, comes with 16K on board. It can easily be upgraded to become a 32K RAM, by the addition of an expansion module which fits inside the ZX-Panda case.

There are no 'mating problems', because the case is contoured to the ZX81 for stability. No wobble problem here. The 16K unit is £25.00, with 16K expansion module at £19.95, and the two together, giving 32K, is £39.95.

Details from Stonechip Electronics, Unit 4, Hoskins Place, Watchetts Road, Camberley, Surrey.

ZX COMPUTING



Britain's Biggest Magazine For The Sinclair User

Make sure you get every issue of *ZX Computing* — Now bi-monthly!

Just £11.50 will ensure the next six issues will be lovingly wrapped and posted to you. Just fill in the form below, cut it out and send it with your cheque or postal order (made payable to ASP Ltd) to:-

ZX Computing Subscriptions
513 London Road,
Thornton Heath,
Surrey CR4 6AR

Alternatively you can pay by Access or Barclaycard in which case simply fill in your card number, sign the form and send it off. Do NOT send your card!

Make the most of your ZX computer with *ZX Computing* — Now bi-monthly!

Subscription Order Form

Cut out and SEND TO:

ZX COMPUTING Subscriptions
513, London Road,
Thornton Heath,
Surrey CR4 6AR

Please commence my subscription to ZX Computing with the very next issue.

SUBSCRIPTION RATES (tick □ as appropriate)

- £11.50 for six issues UK □
- £13 for six issues overseas surface mail □
- £23.80 for six issues overseas airmail □

I am enclosing my (delete as necessary) cheque/Postal Order/International Money Order for £. . . . .
(made payable to ASP Ltd)
OR
Debit my Access/Barclaycard*
(*delete as necessary)

BARCLAYCARD VISA

Please use BLOCK CAPITALS and include post codes.

NAME (Mr/Mrs/Miss) . . . .

ADDRESS . . . .

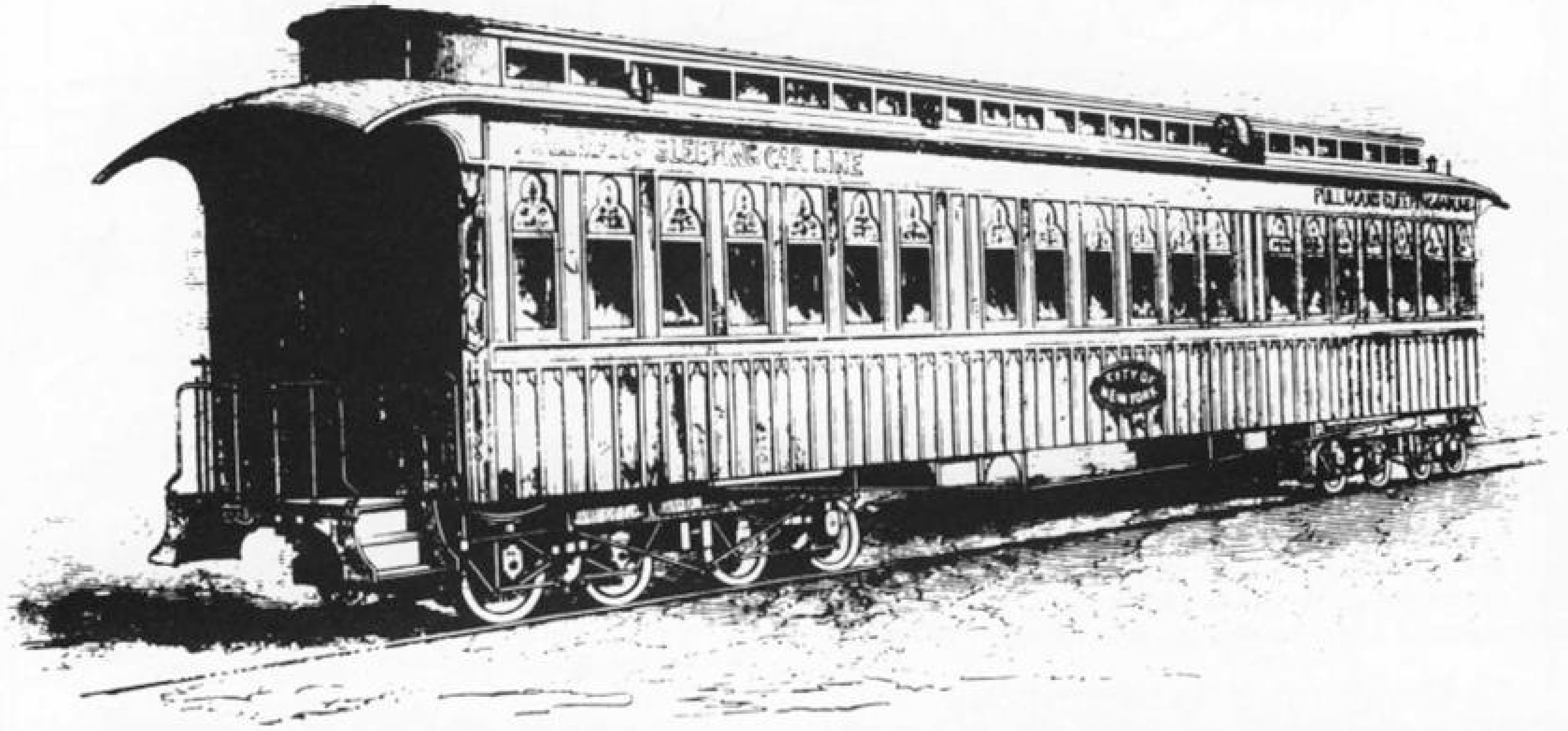
. . . .

. . . . POSTCODE . . . .

Signature . . . .

Date . . . .

Train entries steam in



In the last issue of ZX Computing, we outlined our competition. You were asked to write a 1K ZX81 BASIC program, which had something to do with trains (!), to win either a printer, or one of five sets of the great Psion/Sinclair software.

We were most gratified by the response. Hundreds of entries, many of them showing that a great deal of time and trouble had been spent in creating original programs which, however thinly, had some link with trains.

Winning Program

The first prize, and winner of the ZX Printer, is John Penney of Newton-Le-Willows, Merseyside, for his imaginative program in which you are running along the top of an infinitely long train which is going under a bridge to the right of the screen. You have to run along the track (using the "5" key), and jump over the gaps between carriages (using the "7" key). When you finally misjudge, the number of carriages you've jumped is shown.

Runners-up

The five prizes of sets of Psion's superb software for the ZX81 (which is available from Sinclair Research) go to, (in no particular order of merit):

— Members of the Periton Mead School Computer Club, for their program 'Arithmetic Point Duty'.

— Nick Telfer of Whitehaven, Cumbria, for 'Tay Bridge Disaster', which was suggested by his nine-year-daughter, who owns the ZX81.

— Norman Martin of Kempshott for his 'Signalman' program.

— Paul Caris for his game (which we found extremely difficult to finish with even a tiny profit) to run a railway company; and

— D A Stanford, Kinross, for his program in which you have to keep a steam-driven train going, by feeding coal into the boiler, without blowing up the boiler.

Congratulations, your prizes are on their way to you. Special mention, for programs which nearly made it into the winners' list, for Karri Kaksonen of Finland ('Transportation'), N Howard of Roydon ('Inter-city Tape Hunt'), A Johnson, Seaton Ross ('Track Layer'), T J Allan, Hampton ('Shortest route on the Circle Line') and David Pickering, Bordon ('Build the Line'). Here are the winning programs:

TRAIN ROBBER — John Penney

Run along the train with the "5" key, jump gaps between carriages with the "7". If you get carried to the right of the screen, where the low bridge is, the game ends, and the number of carriages you've jumped is shown.

```
   5 LET C=NOT PI
  10 LET Y=INT PI
  15 LET X=Y
  20 LET A$="████████████████████
██████████"
  25 LET A$="■"+A$( TO 31)
  30 IF RND>.8 AND A$(3)<>" " TH
EN LET A$(2)=" "
  35 PRINT AT Y,X;" "
  40 LET X=X+1-(INKEY$="5")
  45 IF X>30 THEN GOTO VAL "95"
  50 PRINT AT 4,0;A$;AT Y,X;"*"
  55 IF A$(X+1)=" " THEN GOTO VA
L "95"
  60 IF INKEY$="7" THEN GOTO VAL
 "75"
  65 LET C=C+1
  70 GOTO 25
  75■PRINT AT Y,X;" ";AT VAL "2"
,X;"*";AT VAL "2",X;" "
  80 LET A$="██"+A$( TO VAL "30"
)
  90 GOTO VAL "50"
  95 PRINT C
```

STEAM POWER — D A Stanford

You are driving a steam train. There are 60 cwts of coal in the tender and the boiler pressure is 200 psi. You have to travel for 200 miles, adding coal when needed. Too little, and the train will stop; too much and the boiler will explode. Can you make it for 200 miles, without running out of coal?

```
   2 LET C=CODE "W"
   3 LET D=SIN PI
   4 LET P=CODE "COS "
   5 LET V=D
   6 LET F=D
  10 CLS
  15 PRINT "COAL STORE",C,"BOILE
R PRESSURE",P,"DISTANCE GONE",D,
"SPEED",V,,,"ACTION?"
  17 IF P>VAL "300" THEN GOTO PE
EK PI
  18 IF D>CODE "COS " THEN GOTO
PEEK SIN PI
  19 IF P<VAL "150" AND C=SIN PI
 AND V=SIN PI THEN GOTO CODE "CO
S "
  20 INPUT A
  30 IF A<>ABS INT A OR A>PI OR
(A=CODE "■" AND C=SIN PI) THEN G
OTO CODE "="
  40 LET F=F-RND+(CODE "■"*RND A
ND A=CODE "■")
  50 LET C=C-(CODE "■" AND A=COD
E "■")
  60 LET P=P-(V/10)**2+F*F*SGN F
  70 LET V=V+((10 AND V<90) AND
A=2)-((10 AND V>9) AND A=3)
  80 LET V=V AND P>VAL "150"
  90 LET D=D+V/10
 100 GOTO PI*PI
 200 PRINT "NO ";
 211 PRINT "SUCCESS"
 220 GOTO VAL "300"
 255 PRINT "BOILER EXPLODES"
 300 POKE VAL "16410",SGN PI
```

RUN A RAILROAD — Paul Caris

Just enter the number of trains you want to buy, and then how much you'll charge passengers for a 50 mile trip (you get four months to break even, or make money) and see how clever your decisions were.

```
  10 LET A=100000
  20 PRINT "NO. OF TRAINS AT £10
000 ";
  30 INPUT B
  40 PRINT B
  50 LET C=(B*2000)+5000
  60 LET D=B*10000
  70 LET E=A-D
  80 FOR X=1 TO 4
  90 LET E=E-C
 100 LET J=0
 110 PRINT "TICKET PRICE IN £";
 120 INPUT F
 130 PRINT F
 140 FOR Y=1 TO B
 150 LET G=INT (RND*F)+1
 160 LET H=INT ((RND*(3000/G))+(
500/G))
 170 LET J=J+H
```



The prize

```
 180 NEXT Y
 190 PRINT "NO. OF PASSENGERS= "
;J
 200 LET K=F*J
 210 LET E=E+K
 220 IF A<=E THEN PRINT "PROFIT=
£";E-A
 230 IF A>E THEN PRINT "LOSS=£";
A-E
 240 NEXT X
 250 PRINT "ASSETS=£";E
```

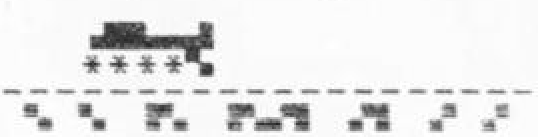
SIGNALMAN — Norman Martin

There are two tracks leading into a tunnel from each end, but only one track is inside the tunnel. You control the entry of trains from the left ("*"), while you have no control over those from the right ("$"). You must never allow a train in the central area while another train is approaching. You win if you get 20 trains safely across.

```
  10 LET U=VAL "1"
  20 LET V=VAL "30"
  30 LET W=VAL "17"
  40 LET X=VAL "20"
  50 LET Y=VAL "0"
  60 LET S=Y
  70 LET T=X
  80 LET R=U
  90 LET D=Y
 100 RAND
 110 PRINT AT Y,Y;"SCORE";AT W+U
,Y;"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀
▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄";AT X-U,28;"▄▄▄▄"
 200 PRINT AT W,V;" ";AT Y,6;S
 210 LET S=S+U
 220 LET L=U
 300 IF L<>3 OR INKEY$<>"" THEN
LET L=L+U
 310 IF D>Y THEN LET D=D-U
 320 IF D<=Y THEN LET R=R-U
 330 PRINT AT W,L-U;" *";AT X,R;
"$ "
 335 IF L>=R AND R>3 AND L<28 TH
EN GOTO 520
 340 IF D>Y THEN GOTO 420
 350 IF R<>U THEN GOTO 380
 360 PRINT AT X,R;" "
 370 LET R=V
 380 IF R<>V-U OR RND<.2 THEN GO
TO 420
 390 LET D=RND*9+24
 400 LET T=T-U
 410 IF T<Y THEN GOTO 520
 420 IF L>V-U THEN GOTO 200
 430 GOTO 300
 520 PRINT AT W,Y;"GAME OVER"
```

TAY BRIDGE DISASTER — Nick Telfer

You have to drive your train (using the "8" key) across the bridge before it collapses. Too slow, and it will crumble beneath your rear wheels, too fast and you'll crash. Good luck. We found this one almost impossibly difficult. Run this program in FAST mode.



```
   5 LET N=0
  10 PRINT AT 10,4;"-----------
---------";AT 11,5;"▚ ▚ ▛ ▛▄▜ ▞
▌ ▞ ▞"
  20 LET D=6
  21 GOSUB 130
  25 FOR T=12 TO 60
  26 IF D=20 THEN GOTO 110
  35 IF INKEY$<>"8" THEN GOTO 41
  36 LET D=D+1
  37 GOSUB 130
  38 IF D>3+60*T/100 THEN GOTO 1
00
  41 PAUSE 50
  45 LET N=INT N+3*RND
  46 IF N<6 THEN GOTO 55
  50 PRINT AT 10,N;"  ";AT 11,N;
" ▘"
  55 IF N>=D THEN GOTO 120
  60 NEXT T
 100 PRINT "TOO FAST"
 101 PAUSE 100
 103 PRINT "YOU CRASHED"
 104 STOP
 110 PRINT "SAVED"
 115 STOP
 120 PAUSE 100
 121 GOTO 103
 130 PRINT AT 8,D;" ▟▀▄▄▙";AT 9,
D;" ****▚"
 140 RETURN
```

ARITHMETIC POINT DUTY — Periton Mead School Computer Club

You are on point duty and must bring the train into the correct platform. You know which station to steer it to (using the "6" and "7" keys) because its number is on the sides of the coaches. The only catch is that you must add them together before it reaches the station.

```
  10 DIM A(5)
  20 LET Y=PI/PI
  30 LET T$=""
  70 LET T$=STR$ INT (RND*(CODE
"█")+Y)+"+"+STR$ INT (RND*(CODE
"█")+Y)
 100 LET D=VAL T$
 110 FOR I=Y TO CODE "█"
 120 LET A(I)=INT (RND*(D+D)+Y)
 130 NEXT I
 140 LET A(INT (RND*CODE "█"+Y))
=D
 150 FOR I=Y TO LEN T$
 160 IF CODE T$(I)<CODE "/" THEN
 NEXT I
 170 LET T$(I)=CHR$ (CODE T$(I)+
CODE "█")
 180 NEXT I
 190 LET T$=T$+"=█"
 192 FOR I=CODE "▒" TO CODE "£"
 195 PRINT AT I,CODE "=";A(I-COD
E "▛")
 197 NEXT I
 200 LET C=CODE "▒"
 205 LET B=C
 210 FOR I=PI/PI TO CODE "="-LEN
T$
 215 PRINT AT B,I-Y;"          "
 220 LET B=C
 230 PRINT AT C,I;T$
 240 LET C=C+(INKEY$="6")-(INKEY
$="7")
 245 IF C>CODE "£" THEN LET C=CO
DE "£"
 247 IF C<CODE "▒" THEN LET C=CO
DE "▒"
 250 FOR J=Y TO CODE "H"
 260 NEXT J
 270 NEXT I
 280 FOR I=Y TO CODE "█"
 290 IF C=I+CODE "▛" AND A(I)=D
THEN GOTO 330
 300 NEXT I
 310 PRINT AT C,CODE "=";"***"
 320 STOP
 330 PRINT AT C,CODE "/";"OK."
```

Controlling Your Cash

A useful selection of business aids for the home and office, including a personal banking system and programs for VAT and book-keeping.



Cash controller

Cash Controller is a new multi-function money program for the Spectrum from Richard Shepherd Software. It combines a personal banking system with home budgeting, and a loan and mortgage calculator. The bank account and budgeting systems are interlinked, so that transactions through the bank account are registered on the budgeting system under headings which are chosen by the user.

The bank account holds up to 200 transactions, and will include standing orders into statements as they become due. The statements can be produced running from any date, to any date, and hard copies can be made using the Sinclair printer.

Better Budgeting

The home budgeting system allows you to estimate expenditure under headings which you choose, and then produces variance figures, showing how close the actual expenditure was to the estimate.

The loan and mortgage calculators will provide figures showing the length of time repayment will take, or the initial sum borrowed, or how much a repayment will be per week or month. Very useful when considering a major financial commitment!

The Password

In a program like this, security is obviously very important. Here, it is ensured by a 'password' which the user chooses, which is only known to him. To avoid any problems with crashing or data scrambling if an incorrect date or figure is entered, this program has been carefully crash proofed. The on screen instructions are supplemented by printed notes.

The Cash Controller costs £10.00, and is available from Richard Shepherd Software, 22 Green Leys, Maidenhead, Berks, SL6 7EZ (0628 21107).

Getting serious

ZX SAS is a new company specialising in Serious Application Software (hence the company's name) for the ZX range of microcomputers.

The initial software offerings from the company include PAYE and Sales Ledger. Programs under development include General and Purchase Ledgers, Retail Accounting, together with some industry-based accounting software and a comprehensive indexing/retrieval system, all for the 16K ZX81. Spectrum versions are under development.

Company director, Barry Ennever, says ZX SAS also offer tailor-made programs for specific individual requirements.

Two powerful segments of the 14K PAYE program offer you options to calculate or check your PAYE code, and using this calcalate your net pay for next week or month.

PAYE CODE CALCULATIONS — this section prompts you with all the allowances you are likely to claim and offers you advice and also alternative sources of extra information if you are uncertain if these allowances are applicable. The program then displays and prints your PAYE code and how it is made up.

NEW PAY CALCULATION — using your PAYE code this is an automatic calculation of National Insurance and PAYE deductions, with options for pension plan and other salary deductions, thus enabling you to produce your next week or month's net pay, even if your PAYE code is changing, you earn extra overtime or your earnings increase, etc.

Both segments include full instructions within the program which is professionally written with full use of the ZX printer, although this piece of equipment is not necessary for full value to be gained from the program.

PAYE costs £6.50, and Sales Ledger is £10.00. ZX SAS are at Orient House, 42/45 New Broad Street, London, EC2M 1QY.

Databank ZX81

Sanderson Software has released an information storage, sorting and retrieval program for the 16K ZX81, called Databank.

The package includes a good ferro C60 cassette tape with the three versions of the Databank program on one side, and a short example containing a record library on the other side, plus a 21 page (size A5) operating manual. Whilst a 21 page manual may seem daunting, in fact a considerable portion of the manual is taken up with explaining the concept of putting office routines on to a computer, a detailed "work-through" of the 'MUSIC' taped program, and several more examples given in some detail.

Each Databank program is in two parts: the main operating program, which is common to all three versions; and the setting-up program with which the user sets up his own layout. With Databank(S), (S for Standard), the setting-up is done "on the screen" using only 3 keys to "DRAW" the required layout. As this takes up about 10% of the memory, versions (V), (for Vocabulary) and (B), (for BASIC) use another method, still easy to follow, which does not take up this memory space.

The layout has the following limits for each data entry: 1 to 3 screen lines; 1 to 9 parts/fields; 1 character per part/field up to a maximum of 29. The program with the smallest data capacity is program (S) at 9000 BYTES, and as the number of entries is automatically related to the size of ENTRY, this is equivalent to over 100 entries with the maximum entry size of 3 lines with one part per line, to 999 with very small entries.

The main operating program has the following major facilities (MENU displayed on the screen): Add entries, each part separately — checked for length, and right or left justified according to letters or numbers, and displayed immediately after pressing NEWLINE.

CORRECT the last ENTRY — provided no other function has been operated after ADD.

FIND, or SEARCH — on any selected part/field by inputting search information.

SORT, — into ascending order of numbers and letters, (numbers first), on any selected part/field.

SAVE, — saves the progam plus DATA.

Databank costs £5.00, and is available mail order only from Sanderson Software, 1 Manor Court, Breaston, DERBY, DE7 3AW.

Income Tax

Micromega, the personal computer dividion of Quantec Systems and Software Ltd., produces a number of serious applications programs for ZX computers, including COMP-U-TAX to work out income tax on the ZX81, and COMP-U-SHARE, a portfolio management program.

Speaking for Quantec, Mr R E Cooke-Hurle said he believed his company was one of the few professional software companies producing 'serious' software for personal computers. The company's other activities include complex commercial software for multiuser installations, and communications software — for example, they are engaged at present on a project for a leading multinational company to link up to 200 geographically dispersed locations, each equipped with a micro into an integrated network all able to intercommunicate.

Mr Cooke-Hurle said: "Our first personal computer product was COMP-U-TAX, which was distributed through W H Smith in considerable volumes. It is perhaps worth recording that the project started off as an 'I wonder if this could be done on a ZX81' type of approach, and it was all done on borrowed equipment."

COMP-U-TAX calculates and checks your income tax bill, has 1981/82 allowances and tax rates built in, and is applicable to UK personal tax payers. The program allows for incomes up to £99999 (so I guess it could cope with yours), will allow for a marital status change during the year, and allows for calculations of a wife's earned income election. The product comes with a well-produced 24 page manual. Micromega are at 230-236 Lavender Hill, London, SW11 1LE (01-223 7672).



Starting off at the sharp end

EXECUTION and ANAGRAMS are two great word games written by Paul Toland for the 16K ZX81. If you haven't got a 16K pack, there's no need to feel left out, as Dilwyn Jones' game CRASHER fits happily within 1K.

Execution

This is a graphic variation on the hangman game. The computer chooses a word randomly from its vocabulary, and you have to guess what the word is, by entering letters one at a time.

You're allowed 10 incorrect guesses before the blade on the guillotine does its job. There is no need to press NEWLINE when entering guesses.

Before you run the program the first time, enter DIM D$(30,10) as a direct command, then start it with GOTO 1 (not RUN). Then, you'll be asked to enter 30 words, one at a time. For subsequent runs, start with GOTO 5, and your vocabulary will be safe. Enter "N" if you do not want another game at the end. SAVE the program by entering GOTO 390. Note that the inverse E in line 390 should be entered as an ordinary E.



Anagrams

The ZX81 randomly selects and scrambles a word. You must repeatedly enter what you think is the correct word, and the computer will underline those letters which are in the correct position.

You have a maximum number of attempts before the computer stops you and reveals the word. You use "N" if you don't want another game, or just press NEWLINE/ENTER if you do.

Entering the initial vocabulary, and the subsequent changing of it, is the same as in Execution. Save the program by entering GOTO 340, and restart the program without losing your vocabulary, by entering GOTO 10. You'll need to set up an array, by entering DIM D$(30,10) in the direct mode before you begin, then start with GOTO 1.

It takes some time to scramble a word, so don't be surprised if nothing happens for the first few seconds of the game.

Crasher

Nothing to do with some computers' habits of blanking out! You are drifting in space, the object being to clear up as much debris as you can, particularly those with a high CODE value. The debris is valued according to the code of the character of debris. The letters score highest, then numbers and so on. Ignore inverse characters if they appear for they do not score because they've been affected by solar radiation and mutated to their inverse form, so are not worth anything. The keys 5 and 8 steer you left and right in the direction of the arrows on the keys. You score when the debris is directly ahead of you on screen so you 'crash' into it. The position of your spaceship on screen is set by X (horizontal position) and Y (vertical position). They are initially set by lines 10 and 20 to a position at about the middle of the playing area of the screen. The variable S records your score and is set to 0 at the start of the game by line 30. F is the variable that controls the duration of the game — it does not count in any particular unit but is a convenient way of controlling the length of the game. The time is clocked up in line 50, one unit of space-time at a time (!) The statement LET F = F + 1 may confuse you a little since it is hard to imagine F being equal to F+1.

Truth

It actually means add 1 to the old value of F to make the new value. Line 60 is rather complicated. It determines where the character is PRINTed, which character is PRINTed and whether it's an inverse character. Let us look at the position first. The character is placed at the bottom of the playing area (the Y co-ordinate is set at 20 and X co-ordinate to a value from 0 to 19 by the random number expression. The expression after the semi-colon generates a character at random from the number generated in brackets after CHR$. The number generated is a random number from 1 to 63. The following expression may look rather strange, but all it does is determine whether to add 0 or 128 to this number (ie. determine whether the character generated is inverse or not. It is a special use of the function AND. What it does is look at the following expression, add 0 to the value if it is not true, or the value before AND if it is true, so that 128 is added to the random

number only if RND is less than .2. This makes it roughly a one in five chance of the character being an inverse, non-scoring character. The statement after this erases the old position of the V before it is scrolled up the screen by line 70. This ensures the V is not PRINTed anywhere other than the middle of the screen. Line 80 checks the keyboard to see whether you're steering to the left, right, or keeping it stationary, and changes the value of X accordingly.

Printing and rushing

The V is PRINTed in its new position in line 90. You can see how the effect given is that the spaceship (V) remains stationary while space seems to rush past. The second part of line 90 moves the PRINT position to immediately ahead of the position of the spaceship. Therefore we may find what is immediately in front of the spaceship by means of the line 100, which finds the CODE of the character stored at that memory location by PEEKing the address held in the system variables 16398 and 16399. Line 110 checks to see if this character ahead of the spaceship by means of the line (ie. it has o CODE of less than 64 rather than less than 128 as you might expect from an inverse character detector. This would permit CHR$ 118 NEWLINE markers that SCROLL might push up the screen. You may like to add this line which will stop the program if you hit a radiation-mutated piece of debris:-

```
115 IF P>127 THEN STOP
```

Execution Listing

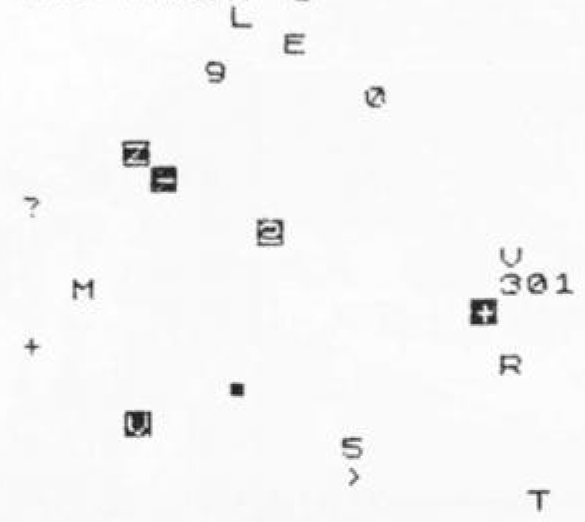
```
   1 FOR I=1 TO 30
   2 INPUT D$(I)
   3 NEXT I
   5 PRINT AT 2,10;"EXECUTION"
   7 PRINT AT 0,0;
  10 RAND
  20 LET G=0
  30 LET C=G
  40 FOR I=1 TO 10
  50 PRINT I;TAB 2;"█   █"
  60 NEXT I
  70 PRINT "   ██████ "
  80 PRINT AT 8,3;"O O"
  90 PRINT AT 9,3;">=<"
 100 LET R=INT (RND*30)+1
 110 LET W$=""
 120 FOR I=1 TO 10
 130 IF D$(R,I)>" " THEN LET W$=
W$+D$(R,I)
 140 NEXT I
 150 LET L=LEN W$
 160 PRINT AT 8,20;"THE WORD"
 165 LET C$="??????????"( TO L)
 170 PRINT AT 10,20;C$;AT 21,5;"
TYPE A LETTER"
 180 PRINT AT G,3;"█████"
 190 IF INKEY$<"A" OR INKEY$>"Z"
 THEN GOTO 190
 195 LET G$=INKEY$
 200 FOR I=1 TO L
 210 IF G$<>W$(I) THEN GOTO 240
 220 LET G=G-1*(G>=0)
 230 LET C$(I)=W$(I)
 240 NEXT I
 250 IF C$=W$ THEN GOTO 320
 260 PRINT AT G,3;"     "
 270 LET G=G+1
 280 IF G<10 THEN GOTO 170
 290 PRINT AT 12,3;"+ +"
 300 PRINT "   -"
 310 PRINT AT 2,20;"COMPLETED"
 315 GOTO 330
 320 PRINT AT 2,20;"SUSPENDED"
 330 PRINT AT 10,20;W$
 340 PRINT AT 21,5;"TRY AGAIN?
"
 350 INPUT A$
 360 IF A$="N" THEN STOP
 370 CLS
 380 GOTO 5
 390 SAVE "EXE█"
 400 GOTO 5
```

Anagrams Listing

```
   1 FOR I=1 TO 30
   2 INPUT D$(I)
   3 NEXT I
  10 RAND
  20 PRINT "       ANAGRAMS"
  30 LET R=INT (RND*30)+1
  40 LET W$=""
  50 FOR I=1 TO 10
  60 IF D$(R,I)>" " THEN LET W$=
W$+D$(R,I)
  70 NEXT I
  80 LET L=LEN W$
  90 LET S$="                    "( TO L)
 100 FOR I=1 TO L
 110 LET R=INT (RND*L)+1
 120 IF S$(R)>" " THEN GOTO 110
 130 LET S$(R)=W$(I)
 140 NEXT I
 150 PRINT "YOUR ANAGRAM IS ";S$
 160 FOR J=1 TO 9
 170 PRINT J;" ";
 180 INPUT G$
 190 PRINT G$
 200 IF G$=W$ THEN GOTO 280
 205 PRINT " ";
 210 FOR I=1 TO LEN G$
 215 IF I>L THEN GOTO 245
 220 IF G$(I)=W$(I) THEN PRINT "
-";
 230 IF G$(I)<>W$(I) THEN PRINT
" ";
 240 NEXT I
 245 PRINT
 250 NEXT J
 260 PRINT "TIME UP - THE WORD I
S ";W$
 270 GOTO 290
 280 PRINT "THAT""S IT - ";
 290 PRINT "TRY AGAIN ?"
 300 INPUT A$
 310 IF A$="N" THEN STOP
 320 CLS
 330 GOTO 10
 340 SAVE "ANA█"
 350 GOTO 10
```

Crasher Listing

```
  10 LET X=10
  20 LET Y=10
  30 LET S=0
  40 LET F=S
  50 LET F=F+1
  60 PRINT AT 20,INT (RND*20);CH
R$ (INT (RND*63+1)+(128 AND RND<
.2));AT Y,X;" "
  70 SCROLL
  80 LET X=X+(INKEY$="8" AND X<1
9)-(INKEY$="5" AND X>0)
  90 PRINT AT Y,X;"V";AT Y+1,X;
 100 LET P=PEEK (PEEK 16398+256*
PEEK 16399)
 110 IF P<64 THEN LET S=S+P
 120 IF F<100 THEN GOTO 50
 130 PRINT S
```





IF YOU WRITE SOFTWARE THEN READ ON...

15 months ago we had written some excellent Games Software for the ZX-81, but when debugging was finally completed we realised that much of the work remained.
Who could we rely on for the effective marketing and production of our efforts?
Who would give our software the chance it deserved?
No company was really giving ZX Software the professional backup and production it deserved so we created Quicksilva to fulfill this role.
Now Quicksilva has developed a set of specific policies to match our beliefs about how a company of this kind should operate.

CUSTOMERS
The highest priority must be given to customer service — "QS customer relations are the best in this field."

SOFTWARE
A high standard must be maintained for all software — "QS reputation for innovative software is firmly established."

PRESENTATION
Should support and enhance the software — "QS policy is to use full colour reproductions of specially commissioned artworks to match the standards of the programs.

ADVERTISING
Needs to be accurate, informative, enjoyable, eye-catching and effective — See recent magazines.

PRODUCTION
Reliable and efficient duplication of cassettes is essential — "QS cassettes are produced by The Tape Duplicating Company Ltd., who have the expertise and equipment to do just that."

DISTRIBUTION
All avenues appropriate to the Software involved must be explored — "QS has over 200 retail outlets in this country alone and a substantial number in all the major overseas markets."

We now know that Quicksilva will give both of us the professional back-up and expertise our programs deserve. It has taken time and effort, and is very much a continually improving service; always we aim for that indefinable 'quality' that makes all the difference. We feel that now is the time to offer 'You' the same service that Quicksilva gives us.
So if you are a Software Author confronted with the same problems we had 15 months ago, let Quicksilva solve them for you and help you as it helped us.

Yours, Nick Lambert & John Hollis

Quicksilva is interested in quality software for the following personal computers — Spectrum — Atari — VIC 20 — B.B.C. — Atom — ZX81 — Electron — Dragon Quicksilva offers you on acceptance of your program:
25% royalties on all sales; the very best professional quality support for your programs;
full credits in advertising, brochures and on cassette for you as the author;
the opportunities and advantages that working with the leaders in the field offers.
Please contact us immediately to discuss this opportunity at

QUICKSILVA, 92 Northam Road, Southampton, Hants SO2 0PB

or ring our Special Authors' Hotline — Southampton (0703) 20169 Ext. 42 — and we will do our best to give you the answer you are looking for.

Snakes alive in outer space

Snakes terrorise humans in the movie Conan the Barbarian, **adding to the general air of mayhem and fear in the film. In these three programs written for the Spectrum by Paul Toland of Derry, Northern Ireland, the beasties to fear are snakes, aliens and hidden landmines.**



Reptile

In this program, you must try to grow your snake as long as possible, by directing it to the pound signs which it eats. However, the £'s remain on the screen for only a short time, after which they turn into poisonous dollar signs (sorry 'bout that, Mr Reagan). The snake will also die if it hits the surrounding border, or itself. Once the game is over, you'll be told how long you became. Note that the A's in line 182 and 285 are graphics A's (that is, A pressed after getting into the graphics mode).

Splodgies from space

In this second program from Paul, your job is to protect the earth from marauding splodgies, which come from the planet Epsilon 1V. The Terran Shield, which protects we poor earthlings from such horrors as the splodgies, has developed a weak spot, and the Epsilonian Mother Ship is hovering just above the weak point. It is bombarding the shield with Splodgie Landers.

You must stop the Splodgies landing and hitting the shield by getting your ship underneath them. If part of the shield is hit, then it will weaken.

If a weak spot is hit, then the shield will collapse, and leave the earth open to attack. You can restore a weak spot by moving your ship over to the spot and pressing 'F'. The shield will collapse if more than nine weak spots are present at any one time. There are graphics characters in lines 70, 80, 150, 160, 200, 230 and 250.

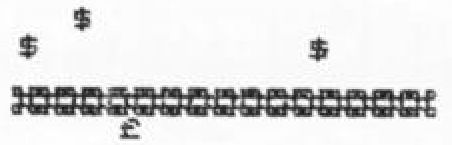
Skilful driving

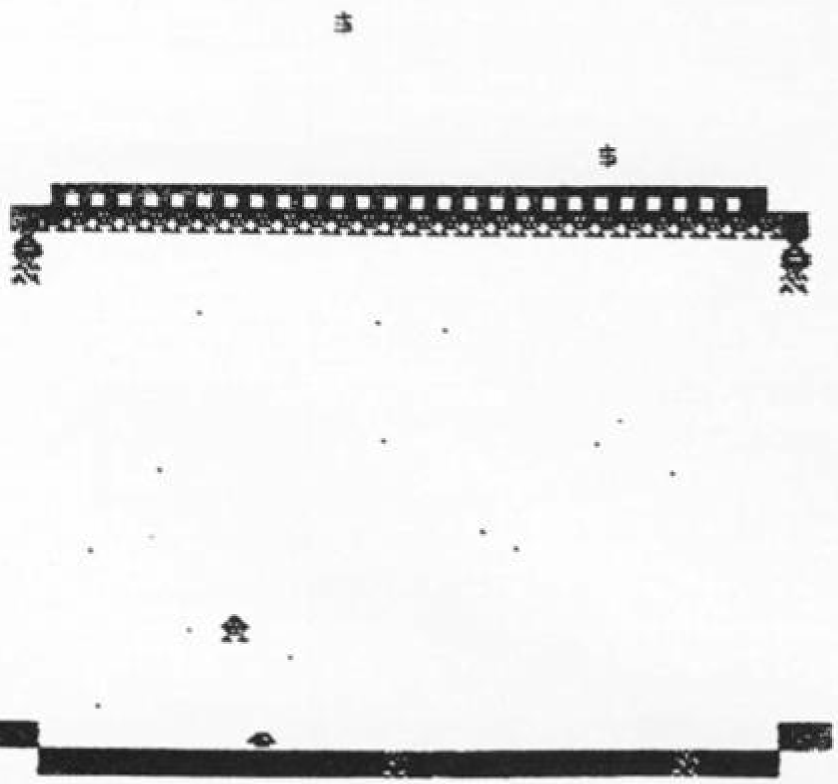
The minefield is full of casualties, and you have to push a wheelchair (!) around the minefield, avoiding the mines and the electrified fence, collect each casualty and bring him or her to hospital (the flashing +). Note that the wheel chair can only carry one person at a time. What a wonderful scenario for a game. User-defined graphics are in lines 100, 160 and 500.

REPTILE

```
   2 GO SUB 400
   5 GO TO 280
   7 LET L=2
   9 RESTORE 480
  10 LET X$=CHR$ 10+CHR$ 11
  20 LET Y$=CHR$ 10+CHR$ 10
  30 INK 6. BORDER 0: PAPER 6: C
LS
  35 FOR I=1 TO 13
  40 READ N: BEEP .2,N
  50 NEXT I
  60 LET MX=INT (RND*32)
  70 LET MY=INT (RND*22)
  80 IF ATTR (MY,MX) <>54 THEN GO
 TO 60
  90 PRINT INK 2;AT my,mx;"£"
 100 BEEP .02,4
 110 FOR i=1 TO 40
 120 LET OX=CODE X$
 130 LET OX=OX+(INKEY$="8")-(INK
EY$="5")
 140 LET OY=CODE Y$
 150 LET OY=OY+(INKEY$="6")-(INK
EY$="7")
 160 IF OX=CODE X$ AND OY=CODE Y
$ THEN LET OX=OX+OX-CODE X$(2):
LET OY=OY+OY-CODE Y$(2)
 165 IF OX<0 OR OX>31 OR OY<0 OR
 OY>21 THEN GO TO 240
 170 IF ATTR (OY,OX)=50 THEN LET
 L=L+1
 180 IF ATTR (OY,OX)=52 THEN GO
TO 260
 182 PRINT INK 4;AT OY,OX;"A"
 185 PRINT AT CODE Y$(L),CODE X$
(L);" "
 190 LET X$=CHR$ OX+X$( TO L)
 200 LET Y$=CHR$ OY+Y$( TO L)
 210 NEXT I
 220 IF ATTR (MY,MX)=50 THEN PRI
NT INK 4;AT MY,MX;"$"
 230 GO TO 60
 240 PRINT INK 6; PAPER 1;AT 0,0
;"YOU CRASHED INTO THE SURROUNDI
```

```
NGWALL-----";
 250 GO TO 270
 260 PRINT PAPER 1; INK 6;AT 0,0
;"YOU HIT SOMETHING YOU SHOULD N
OT HAVE-";
 270 PRINT PAPER 1; INK 6;"AFTER
 GROWING TO ";L
 275 BEEP 3,-10
 280 PAPER 1: INK 6
 285 PRINT '"AAAAAAASNAKESAAAAAS
NAKESAAAAAAAAA"
 290 PRINT '"YOU ARE DIRECTNG A
MONEY SNAKE  AROUND THE SCREEN U
SING THE KEYS 5,6,7&8.   YOUR AI
M IS TO MAKE THE SNAKE GROW BY G
UIDING IT TO THE £ SIGNS,ON WHIC
H IT FEEDS."
 300 PRINT '''" EACH £ REMAINS ON
 THE SCREEN   FOR A SHORT PERIOD
 BEFORE        CONVERTING INTO A
$ WHICH KILLS IF EATEN. THE SNAK
E ALSO DIES    IF IT BITES ITSELF
'''" PRESS 'Y' TO START THE GAME
 OR 'N'TO STOP "
 310 IF INKEY$="" THEN GO TO 310
 320 IF INKEY$="Y" THEN GO TO 7
 340 IF INKEY$="N" THEN STOP
 350 GO TO 310
 400 REM CHR$ GENERATOR
 410 FOR I=0 TO 7
 420 READ N
 430 POKE USR "A"+I,N
 440 NEXT I
 450 DATA BIN 11100111,BIN 10100
101,BIN 11111111,BIN 00100100,BI
N 00100100,BIN 11111111,BIN 1010
0101,BIN 11100111
 460 RETURN
 470 REM MUSIC DATA(HISSING SID?
)
 480 DATA 0,0,4,7,12,11,9,-1,-1,
2,11,9,7
```





SPLODGIES FROM SPACE

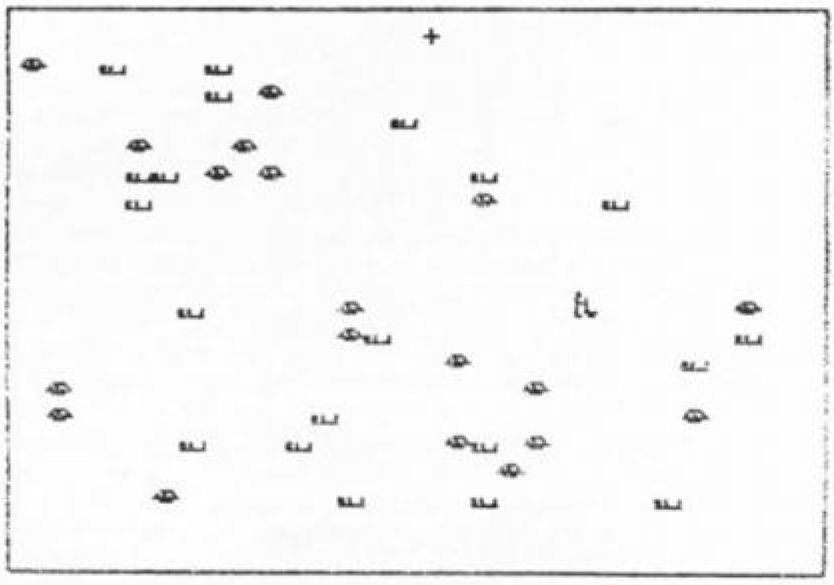
```
   5 GO SUB 400
   7 GO TO 310
  10 PAPER 1: BORDER 1: CLS : IN
K 6: OVER 0
  12 CLS
  15 FOR I=1 TO 20: PLOT RND*254
,RND*174
  17 NEXT I
  20 PRINT AT 20,0;"██";TAB 30;"
██";AT 21,3;"SKILL LEVEL 1-5. 5=
HARDEST"
  30 LET I$=INKEY$
  40 IF I$<"1" OR I$>"5" THEN GO
 TO 30
  50 LET SKILL=VAL I$
  55 INK 3
  60 PRINT AT 21,2;"██████████████
█████████████████"
  70 PRINT AT 0+SKILL-1,2; INK 7
; PAPER 2;"██████████████████████
███████"; PAPER 1;"  ";"█AAAAAA
AAAAAAAAAAAAAAAAAAAAA█  B";TAB
30;"B  "; FLASH 1; INK 2;"C"; FL
ASH 0;TAB 30; FLASH 1;"C"
  75 INK 4
  80 PRINT AT SKILL,2; OVER 1;"D
DDDDDDDDDDDDDDDDDDDDDDDDDDD"
  90 LET HIT=0: LET TIME=0
 100 LET SHIP=16: OVER 1
 110 LET LAND=INT (RND*28)+2
 120 IF HIT=9 THEN GO TO 290
 125 LET DUR=.03-TIME/8000
 130 LET TIME=TIME+1
 140 FOR I=SKILL+1 TO 20
 150 PRINT AT I-1,LAND;"D";AT I,
LAND;"D"
 160 IF I=11 THEN PRINT AT SKILL
,LAND;"D"
 170 PRINT AT 20,SHIP; OVER 0;"
"
 175 BEEP DUR,22-I
 180 LET SHIP=SHIP+(INKEY$="8")-
(INKEY$="5")
 190 LET SHIP=SHIP+(SHIP=1)-(SHI
P=30)
 200 PRINT AT 20,SHIP; OVER 0;"E
"
 210 IF INKEY$="F" AND ATTR (21,
SHIP)<>11 THEN PRINT INK 3; OVER
 0;AT 21,SHIP;"█": LET HIT=HIT-1
 220 NEXT I
 230 PRINT AT 20,LAND;"D"
 240 IF LAND=SHIP THEN GO TO 110
 250 IF ATTR (21,LAND)=11 THEN P
RINT AT 21,LAND; INK 7;"C": LET
HIT=HIT+1: BEEP .1,-20: GO TO 11
0
 260 OVER 0
 270 PRINT AT 0,0;"THE FORCE FIE
LD HAS BEEN BROKEN "
 280 GO TO 300
 290 OVER 0
 295 PRINT AT 0,0;"THE SHIELD CO
LLAPSED UNDER THE  STRAIN OF SO
MANY WEAK POINTS          "
 300 BEEP .2,-20: BEEP .9,-10
 305 PRINT "YOUR SCORE IS ";TIME
*SKILL
 310 PRINT '''"YOU HAVE BEEN CHOS
EN TO DEFEND  AN AREA OF THE EAR
TH'S DEFENCE  SHIELD WHICH HAS W
EAKENED.        THE ALIENS HAVE D
ISCOVERED THISWEAKNESS AND WILL
TRY TO BREAK  THE SHIELD BY DROP
PING ON TO IT,PROTECT THE SHIELD
 BY GETTING    UNDERNEATH THE ALI
EN.            ONE HIT WILL WEAKE
N THE FIELD.   AND A HIT ON A WEA
K SPOT WILL    BREAK THE FIELD. Y
OU CAN RESTOREA WEAK SPOT DIRECT
LY BELOW BY    PRESSING 'F', THE
SHIELD WILL    SHATTER IF THERE A
RE 10 WEAK     SPOTS."
```

```
 320 PRINT "PRESS 'Y' TO START O
R 'N' TO END"
 330 LET I$=INKEY$
 340 IF I$="Y" THEN GO TO 10
 350 IF I$="N" THEN STOP
 360 GO TO 330
 399 STOP
 400 READ A$: IF A$="Z" THEN RET
URN
 410 FOR I=0 TO 7
 420 READ N: POKE USR A$+I,N
 430 NEXT I
 440 GO TO 400
 450 DATA "A",BIN 11100111,BIN 1
1000011,BIN 10000001,0,0,0,0,0
 460 DATA "B",BIN 00111100,BIN 0
0011000,BIN 00111100,BIN 0110011
0,BIN 11111111,BIN 11000011,BIN
11111111,BIN 11111111
 470 DATA "C",BIN 01010100,BIN 0
0101000,BIN 01001010,BIN 1011010
1,BIN 00010000,BIN 01001010,BIN
11100110,BIN 10010001
 480 DATA "D",BIN 00011000,BIN 0
0111100,BIN 01010110,BIN 1111111
1,BIN 01101010,BIN 00111100,BIN
01000010,BIN 11100111
 490 DATA "E",0,0,0,BIN 00111100
,BIN 01010010,BIN 11111111,BIN 0
0111100,0
 500 DATA "Z"
```

SKILFUL DRIVING

```
  20 GO SUB 400
  22 GO TO 610
  25 INK 6
  30 PAPER 6: BORDER 6: CLS
  40 INK 2
  50 PLOT 4,4: DRAW 247,0
  60 DRAW 0,167: DRAW -247,0
  70 DRAW 0,-167
  80 INK 0
  90 FOR I=1 TO 20
 100 PRINT AT RND*17+2,RND*28+1;
"D"
 110 NEXT I
 115 INK 1
 120 FOR I=1 TO 20
 130 LET X=INT (RND*28)+1
 140 LET Y=INT (RND*17)+2
 150 IF ATTR (Y,X)<>54 THEN GO T
O 130
 160 PRINT AT Y,X;"C"
 170 NEXT I
 190 INK 3: OVER 0
 200 LET CAR=0
 210 LET RET=0
 220 LET X=16: LET Y=1
 222 LET A=0: LET D=0
 230 PRINT AT Y,X;" "
 235 PRINT AT 1,16; FLASH 1; INK
3;"+"
 237 LET I$=INKEY$
 240 IF I$="5" THEN LET A=-1: LE
T D=0
 245 IF I$="8" THEN LET A=1: LET
D=0
 250 IF I$="6" THEN LET A=0: LET
D=1
 255 IF I$="7" THEN LET A=0: LET
D=-1
 257 LET X=X+A: LET Y=Y+D
 260 LET CH=ATTR (Y,X)
 265 PRINT AT Y,X;CHR$ (144+CAR)
 270 IF CH=48 THEN GO TO 500
 280 IF CH=50 THEN GO TO 520
 290 IF CH=49 AND CAR THEN GO TO
540
 300 IF CH=49 THEN LET CAR=1
 310 IF CH=179 AND CAR THEN LET
CAR=0: BEEP .2,5: LET RET=RET+1:
 IF RET=20 THEN GO TO 560
 320 PAUSE DELAY
 330 GO TO 230
 400 READ A$: IF A$="Z" THEN RET
URN
 410 FOR I=0 TO 7
 420 READ N: POKE USR A$+I,N
 430 NEXT I
 440 GO TO 400
 450 DATA "A",BIN 11000000,BIN 1
1000000,BIN 10010000,BIN 1111000
0,BIN 10010000,BIN 10010000,BIN
10011110,BIN 11001100
 460 DATA "B",BIN 11000000,BIN 1
1000000,BIN 10011100,BIN 1111110
0,BIN 10011000,BIN 10011111,BIN
10011111,BIN 11001101

 470 DATA "C",0,0,0,0,0,BIN 1101
0001,BIN 11010001,BIN 11111111
 480 DATA "D",0,0,0,BIN 00111100
,BIN 01010010,BIN 11111111,BIN 0
0111100,0
 490 DATA "Z"
 500 PRINT AT Y,X; OVER 1; FLASH
1;"*": BEEP 3,-30
 510 PRINT AT 1,1;"YOU HIT A MIN
E": GO TO 600
 520 PRINT AT 1,4; FLASH 1; INK
2;"ZZZZZZZZZZAAAAAPPPPP"
 525 FOR I=-30 TO 0 STEP 3
 530 BEEP .2,I: BEEP .02,I+10
 535 NEXT I: GO TO 600
 540 PRINT AT 1,2;"THE WHEELCHAI
R IS OVER-LOADED": GO TO 600
 560 BEEP 1,5: BEEP 3,8
 570 PRINT "YOU DONE IT!!!"
 600 INK 0
 605 PRINT " YOU RESCUED ";RET;"
OUT OF THE 20"
 610 PRINT ''"DDDDDDDDDMINE FIEL
DDDDDDDDDDDDD"
 620 PRINT '"THERE ARE 20 CASUAL
TIES LYING    IN THE MINEFIELD. I
T IS YOUR JOBTO BRING THEM TO TH
E HOSPITAL     IN A WHEELCHAIR ONE
 AT A TIME.  THE FENCE AROUND TH
E FIELD IS    ELECTRIFIED SO AVOI
D CONTACT     WITH IT."
 630 INPUT "CHOOSE A SKILL LEVEL
 0 TO 5      (5=THE EASIEST)";DEL
AY
 640 IF DELAY<0 OR DELAY>5 THEN
GO TO 630
 650 LET DELAY=DELAY*2+1
 660 RESTORE 660
 670 FOR I=1 TO 16
 680 READ N: BEEP .2,N
 690 NEXT I
 700 GO TO 25
 710 DATA 3,1,0,1,0,1,0,-2,-2,0,
-2,0,-2,0,-2,-4
```



Number crunching on the Spectrum and ZX81

Dr Frank O'Hara has a look at what is really going on inside the ZX computers while they are carrying out arithmetic operations. He says he has discovered some programming weaknesses in the ROM, and even 'the occasional outright blunder'.

A good deal has been written about the problems of doing arithmetic on the ZX81. There is the general point that computers do not hold all decimal numbers exactly, and the further limitation of the ZX81 ROM to 8 kilobytes, not all devoted to arithmetic. There are also some programming weaknesses, and even the occasional outright blunder.

It occurred to me that it might be useful to have a BASIC program to show exactly what decimal number the ZX81 or Spectrum is holding, even when it is not the number you thought it was holding! This is possible because a 5 byte floating point number can be expressed exactly in decimal, even though it may take a lot of digits to do this. The result of my efforts is Program 3. Because it is long and complex, it seems best to introduce the subject by means of two shorter programs which already clarify a lot of the arithmetic.

Program 1 has two versions. Line 20 is the general version and line 30 is a special application. Line 20 shows you exactly how any given number of expression is being held on the ZX81 or Spectrum, as a 5 byte floating point number. For the ZX81, 23627 must be replaced by 16400 and 23628 by 16401; there must of course be only one statement per line and the INPUT quotes must be replaced by PRINT quotes and then INPUT X$. Note that the assignment of a value to v has been left inside the i loop so that the program will work on the 1K ZX81.

Results

Run the general version in line 20 and it will give you quite a lot of interesting results. For example, the expression ½ gives 128 0 0 0 0. It is easy to see from the manual that this is correct. But .5 gives 127 127 255 255 255, one bit short. This will need investigation. Continuing for the present with program 1, .1 or 1/10 gives 125 76 204 204 204. Those 204's seem to cry out for rounding up, don't they? Sure enough, there is a blunder here in the ROM, still present in the Spectrum. Another example of this gremlin is the expression 1/3. This gives 127 42 170 170 170 and you get no prize for guessing that the last 170 should have been rounded up to 171. How about trying some integers, I hear you ask. Well, 10 gives 0 0 10 0 0 on the Spectrum but 132 32 0 0 0 on the ZX81. That is because the Spectrum has a special way of holding integers which are smaller than 65536, leading to greater speed of operation and so restoring some of the advantages of the ZX80. These small integers will not give us any trouble anyway, so we will move on to larger ones.

Entering 1e5 gives 145 67 80 0 0 and 1e13 gives 172 17 132 231 42. Those bytes may not mean a lot to you now, but one object of program 3 will be to show that they are exactly correct. Let me state it plainly: all the powers of 10 from 1e1 to 1e13 are held exactly on the ZX81 and Spectrum. The first imprecision arises with 1e14, as well shall see below. This is very useful for multiple precision BASIC, and it is a lot more than the manual promised. Let us not miss the virtues of these machines in our eagerness to catch Uncle Clive napping! But beware of expressions like 10n (10**n on the ZX81). Even 101 is not precise. Try it and see. It gives 132 32 0 0 1 which you will see is one bit over what 10 gave on the ZX81. For precision, 1e7, say, is always preferable to 10 7. You may need to use 10n in a program because the expression 1en is not allowed; but it will be no use to you if you need an exact integer result. Rounding it might work, but should be checked first.

Finally, try a few expressions like 2n. I give you two inexact ones here. 213 gives 142 0 0 0 6. That last 6 is the error. On a ZX81 you can check direct that 8192 gives 142 0 0 0 0. This is easy enough to check from the manual anyway. On the Spectrum, 8192 is of course a short integer. So here is a longer one that can be checked on the Spectrum too. Enter 226 and you will get 155 0 0 0 11. The 11 is the error term. To find 2 to the 26th, you can exit from program 1 (press EDIT and then STOP) and enter the command PRINT 226 (or PRINT 8192*8192). This gives 67108864. Run program 1 again and enter 67108864 digit by digit. Sure enough, it gives 155 0 0 0 0, confirming that the 11 in the last byte was wrong.

The integral powers of 2 are so important that you may like to investigate them fully. The second version of program 1, in line 30, enables you to do that. The program can be derived from line 20 by editing the first part of it. When run, it shows clearly the 19 cases between 0 and 126 in which the expression 2n does not

give an exact result. Similar inaccuracies occur at the corresponding negative powers, as you can easily check. If you are interested in doing exact numerical work, it can be useful to know what these 19 cases are, and how you can get them exactly and easily. For example, Euler's famous prime number, 231-1, can be got exactly by either 2*230-1 or 232/2-1.

Decimal Fractions

Before moving on to program 3 it will be useful to look at a small program which enables those decimal fractions like .5 and .1 to be correctly produced on the ZX81 and Spectrum. Program 2 is restricted to numbers which start with a decimal point and contain no e-format. It converts such numbers to floating point form more accurately than the existing ROM routine, and it rounds them up as needed too. Entering .5 now gives 128 0 0 0 0 as it should; .25 gives 127 0 0 0 0, .125 gives 126 0 0 0 0, and so on. In addition .1 gives 125 76 204 204 205, correctly rounded up. Entering .3333333333333333 (sixteen 3's) gives 127 42 170 170 171, correctly rounded up. It is worth mentioning here that the reason why .1 and .5 are incorrect on the ZX81 and the Spectrum is that the division routine jumps to the wrong place in trying to get bit 34, and that this could be corrected by changing one byte in each ROM.

Positive Numbers

And so we come to program 3. This is a multiple precision BASIC program which converts a 5-byte floating point number exactly to decimal form. It deals only with positive numbers. Negative numbers do not show any special features here. It does not deal with "short integers" (less than 65536) as they occur on the Spectrum. They too present no problems. It could be typed into a ZX81 mainly by using a new line for each statement. I aim here to show you how it works. Then you can use it for any numbers that interest you.

Division

The heart of the program is in line 50, which stores the constants needed to divide by the first 4 powers of 256, in multiple precision form. The division is done by multiplication, and the result is exact. This can give up to 39 digits before the decimal and up to 159 digits after it. No attempt is made to truncate here. You can round off as you wish; as the program stands, it tells you the whole story.

Decimal Numbers

Three types of result are possible: a decimal greater than ½ and less than 1, a larger number and a smaller fraction. Enter a number or an expression between ½ and 1, and the program returns in about 7 seconds the exact decimal equivalent of the 5 bytes which the machine holds for that number. For example, .75 gives the exact result 0.75 followed by 34 zeros. There have to be 32 decimal places to give exact results; we are working in sixes, so there are 36, the last 4 always zero. The program also gives the actual 5 bytes held by the machine, here 129 192 0 0 0. Note that I have left the second byte as a true numerical byte since all numbers here are positive, but if you want to alter the program to subtract 128 from it feel free to do so. Try also .8 at this stage. This gives an inexact result. You will see that it starts with .799 . . ., that the inaccurary begins at the tenth decimal place, and that 30 decimal places are significant (not 32, because the last byte is divisible by 4). Finally, in this range try .91 to see a result which has 32 significant figures after the decimal.

The second type of result is the larger number. This takes up about 30 seconds for about 1e38 (or about 2 to the 126). Line 1400 has been coded to speed the original draft up by a factor of about 3, but those digits take time to produce (and the exponent takes time to insert). Here you can test those powers of 10 from 1e5 to 1e13 and see that they are all exact. (1e13 takes 16 seconds). It is interesting to see that the error in 1e14 is 16384. Since the exponent is 2 to the 47, the 32nd bit can only be correct to the nearest 2 to the 15 i.e. 372768, and this leads to an error of plus 16384. Of course this is just as it should be, and shows the system really working. It is also of interest at this point to look at the largest power of 10 that fits into the machine, 1e38, though the result may not tell you a lot.

The third type of result is the longest, with up to 159 figures after the decimal, with 3 final zeros always shown. These numbers take from about 25 to 52 seconds; producing those digits takes time! .5 belongs to this group; it starts with .499 . . ., is wrong from about the tenth place, and has 33 figures after the decimal (only one division by 2 was needed for the exponent). At the other extreme, 1e-38 takes 52 seconds, shows a similar scale of inaccuracy, and gives 157 figures after the decimal (not 159 because the last byte is divisible by 4).

Finally

Program 3 also provides for the input of 5-byte floating point numbers in line 25. To use this, add the two lines
15 GO TO 25
62 GO TO 70
and change line 180 to GO TO 15 (to restore the program, delete lines 15 and 62 and change line 180 back to GO TO 20). Note that I am still using the true numerical second byte, and if you want to alter that you will need to add LET m(1)=m(1)+128 at the end of line 25 and LET m(1)=m(1) −128 at the beginning of line 70.

Perhaps the main numbers of interest for this second method of input are the largest and smallest: i.e. 255 255 255 255 255 gives the full 39 digits of the integer 2127 − 295, the largest integer which the ZX81 and Spectrum can hold. You might not guess from the manuals that this number is held exactly on the machine! At the other extreme, the smallest number is of course 2−128, entered by 1 128 0 0 0; but perhaps a more interesting number is 2−127 − 2-159, entered by 1 255 255 255 255, also taking 52 seconds, and showing the maximum of 159 significant figures after the decimal. You may also like to use this method of input to enter some of the numbers of special interest revealed by programs 1 and 2 above.

Machine Code

Program 4 is a machine code version of Program 3. The arithmetic has been done in binary coded decimal, and most of the printing has been done in machine code too. Hence the longest number now appears in less than a second. This should greatly extend the usefulness of the program. There is string input too, (line 30) so the program can deal with any expression, function or variable, not just a number. There is provision for negative numbers, and for short integers as held on the Spectrum. (To see the full five byte floating after the decimal point, add line 3 POKE 30722, 27 before running. This gives 54 places after the decimal for small fractions. To restore the full 159, delete line 3 and run again. For the alternative 5 byte input, add line 25 GO TO 40. To return to normal input, delete line 25. Note that the program does not check whether your five bytes form a legitimate number. After a break, re-enter at line 30.

To see a Spectrum bug, enter "INT-65536"; you may be surprised at the answer! In one respect this program does better than the unaided Spectrum. Expressions like "-65535-1" (or "-65000-536" etc.) give the correct answer here. On the unaided Spectrum they give − 1e-38!

Summing-up

I hope this article goes a long way to dispelling any mystery about how numbers are held on the ZX81 and Spectrum. Although Program 3 is long and complex, I believe that it is a mine of information (I trust that phrase is not copyright!) and can be used far beyond the cases that I have looked at or indicated here.

Program 1

```
  10 REM "5 bytes"
  20 LET x=0: INPUT "Enter a num
ber or expression ";x$: LET x=VA
L x$: PRINT x$;" = ";: FOR i=1 T
O 5: LET v=PEEK 23627+256*PEEK 2
3628: PRINT PEEK (v+i);" ";: NEX
T i: PRINT : GO TO 20
  30 LET x=1: FOR j=0 TO 126: LE
T x=2↑j: PRINT "2↑";j;" = ";: FO
R i=1 TO 5: LET v=PEEK 23627+256
*PEEK 23628: PRINT PEEK (v+i);"
";: NEXT i: PRINT : NEXT j
```

Program 2

```
3000 REM "decimal"
3010 LET a=0: LET v=PEEK 23627+2
56*PEEK 23628
3020 INPUT "Enter a number that
starts with a decimal point and
contains no E-format ";a$
3030 LET b=VAL a$(2 TO ): LET d=
1: FOR i=1 TO LEN a$-1: LET d=10
*d: NEXT i: LET a=b/d
3040 IF b-a*d)=d*2↑(PEEK (v+1)-1
51) THEN POKE v+5,PEEK (v+5)+1
3050 PRINT a$;" = ";: FOR i=1 TO
 5: PRINT PEEK (v+i);" ";: NEXT
i: PRINT : GO TO 20
```

Program 3

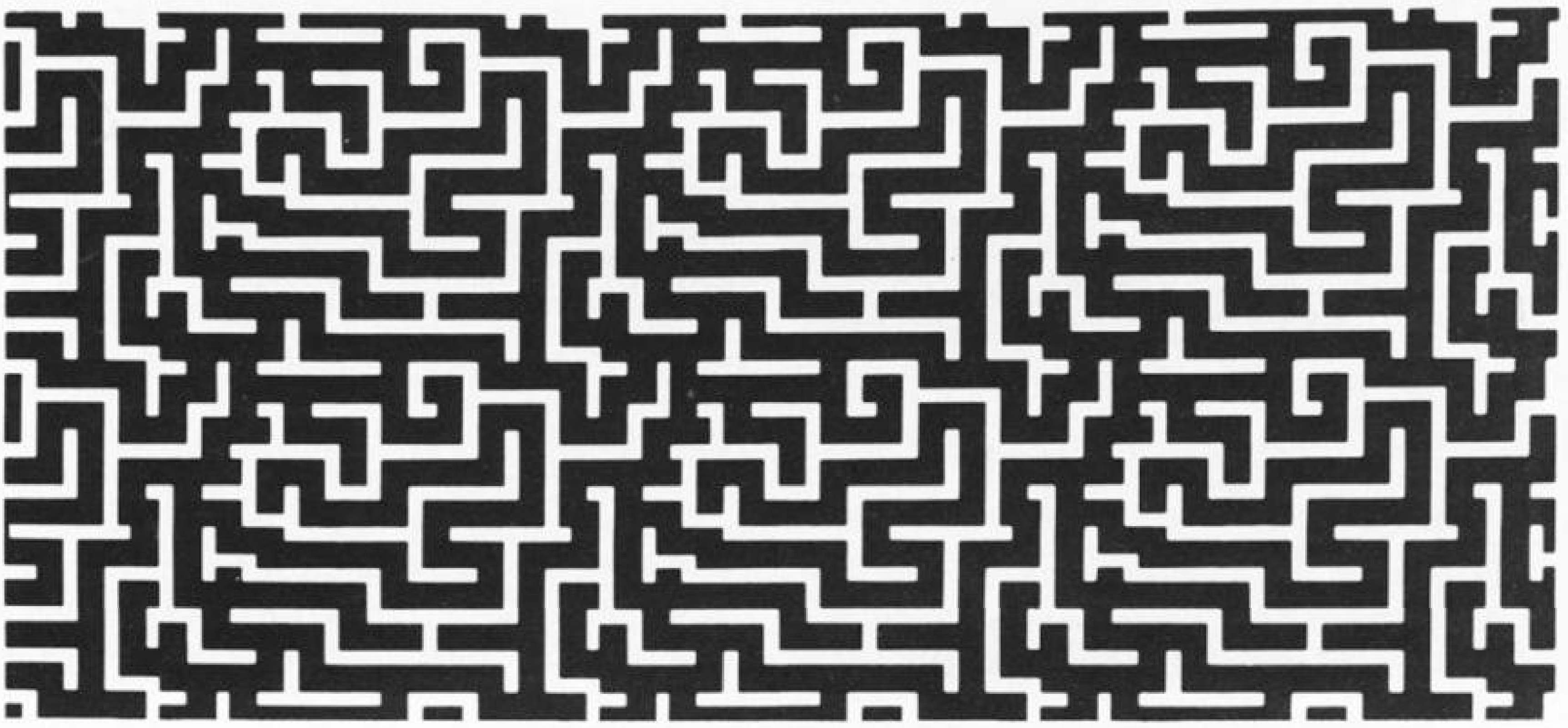
```
  10 REM "number"
  20 INPUT "Enter a positive nu
mber ";x: LET v=PEEK 23627+256*P
EEK 23628: LET e=PEEK (v+1): DIM
 m(4): FOR i=1 TO 4: LET m(i)=PE
EK (v+i+1): NEXT i: LET m(1)=m(1
)+128: GO TO 30
  25 DIM m(4): INPUT e;" ";m(1);
" ";m(2);" ";m(3);" ";m(4)
  30 DIM r(34)
  40 DIM c(4,6)
  50 LET c(1,1)=3906: LET c(1,2)
=250000: LET c(2,1)=15: LET c(2,
2)=258769: LET c(2,3)=62500: LET
 c(3,2)=59604: LET c(3,3)=644775
: LET c(3,4)=390625: LET c(4,2)=
232: LET c(4,3)=830643: LET c(4,
4)=653869: LET c(4,5)=628906: LE
T c(4,6)=250000
  60 FOR k=1 TO 4: LET b=m(k): D
IM a(6): FOR j=1 TO 6: LET a(j)=
c(k,j): NEXT j: GO SUB 1000: NEX
T k
  65 PRINT x;": ";
  70 PRINT e;" ";m(1);" ";m(2);"
 ";m(3);" ";m(4);" = ";
  80 LET r$="": IF e<=128 THEN L
ET r$="0"
  90 IF e<128 THEN GO TO 160
 100 IF e=128 THEN GO TO 150
 110 LET e=e-128: LET s=1100: GO
 SUB 1400
 120 FOR i=1 TO 7: IF r(i)<>0 TH
EN LET r$=r$+STR$ r(i): GO TO 14
0
 130 NEXT i
 140 FOR j=i+1 TO 7: GO SUB 1300
 NEXT j
 150 LET r$=r$+".": FOR j=8 TO 1
3: GO SUB 1300: NEXT j: GO TO 17
0
 160 LET r$=r$+".": LET e=128-e:
 LET s=1200: GO SUB 1400: FOR j=
8 TO 34: GO SUB 1300: NEXT j
 170 PRINT r$
 180 GO TO 20
1000 FOR i=8 TO 13: LET r(i)=r(i
)+a(i-7)*b: NEXT i
1010 FOR i=13 TO 9 STEP -1: GO S
UB 2000: NEXT i
1020 RETURN
1100 FOR i=1 TO 13: LET r(i)=d*r
(i): NEXT i
1110 FOR i=13 TO 2 STEP -1: GO S
UB 2000: NEXT i
1120 RETURN
1200 LET h=0: FOR i=8 TO 34: LET
 h=h-d*INT (h/d): LET r(i)=h*1e6
+r(i): LET h=r(i): LET r(i)=INT
(h/d): NEXT i: RETURN
1300 FOR k=1 TO 5: IF r(j)<10↑k-
.5 THEN LET r$=r$+"0"
1310 NEXT k: LET r$=r$+STR$ r(j)
: RETURN
1400 LET f=INT (e/8): LET g=e-8*
f: LET d=256: FOR j=1 TO f: GO S
UB s: NEXT j: LET d=2: FOR j=1 T
O g: GO SUB s: NEXT j: RETURN
2000 LET r(i-1)=r(i-1)+INT (r(i)
/1e6): LET r(i)=r(i)-1e6*INT (r(
i)/1e6): RETURN
```

Program 4

```
   1 LET x=0
   2 POKE 30720,14: POKE 30721,1
20: POKE 30722,80: LET a=30750
   4 RESTORE : FOR i=0 TO 63: RE
AD n: POKE a+i,n: NEXT i
   6 DATA 0,57,6,37,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,21,37,135,137,6,
37,0,0,0,0,0,0,0,0,0,0,0,5,150,4
,100,71,117,57,6,37,0,0,0,0,0,0,
0,0,2,50,131,6,67,101,56,105,98,
137,6,37
   8 LET a=30976
  10 FOR i=0 TO 260: READ n: POK
E a+i,n: NEXT i
  12 DATA 6,8,79,203,25,197,48,1
7,6,16,17,145,120,33,109,120,167
,26,142,39,18,27,43,16,248,6,16,
33,109,120,167,126,143,39,119,43
,16,249,193,16,218,201,6,100,33,
110,120,54,0,35,16,251,33,0,120,
54,14,6,4,42,75,92
  13 DATA 35,197,35,229,126,203,
80,40,2,198,128,42,0,120,17,16,0
,25,34,0,120,17,94,120,1,16,0,23
7,176,205,0,121,225,193,16,221,4
2,75,92,35,126,254,129,245,56,20
,214,128,71,197,6,36,33,145,120,
167,126,143
  14 DATA 39,119,43,16,249,193,1
5,239,6,36,33,145,120,126,230,24
0,203,63,87,203,63,203,63,130,86
,119,122,230,15,134,119,43,16,23
5,241,214,128,48,24,237,68,71,33
,130,120,167,197,6,80,126,48,2,1
98,100,203,63,119,35,16,245,193,
16,235
  16 DATA 42,75,92,35,126,254,12
9,245,56,31,6,20,33,110,120,126,
167,35,32,2,16,249,229,197,205,4
0,45,205,227,45,193,225,5,40,11,
205,4,122,35,16,250,24,3,62,48,2
15,62,46,215,241,254,128,33,130,
120,56,9,6,16,205,4,122,35,16,25
0,201,56,2,120,71,205,4,122,35,1
5,250,201,229,197,126,254,10,48,
5,245,62,48,215,241,205,40,45,20
5,227,45,193,225,201
  20 LET x=0: LET v=PEEK 23627+2
56*PEEK 23628
  30 INPUT "Enter a number or e
xpression ";x$: LET x=VAL x$: GO
 TO 50
  40 DIM a(5): INPUT a(1);" ";a(
2);" ";a(3);" ";a(4);" ";a(5): F
OR i=1 TO 5: POKE v+i,a(i): NEXT
 i: PRINT '': GO TO 60
  50 PRINT ''x$;": ";
  60 FOR i=1 TO 5: PRINT PEEK (v
+i);" ";: NEXT i: PRINT "= ";
  70 IF PEEK (v+1)<>0 THEN GO TO
 100
  80 LET r=PEEK (v+3)+256*PEEK (
v+4): IF PEEK (v+2)=255 THEN LET
 r=r-65536
  90 PRINT r: GO TO 25
 100 IF PEEK (v+2)>=128 THEN PRI
NT "-";: POKE v+2,PEEK (v+2)-128
 110 RANDOMIZE USR 31018: GO TO
25
```

Getting stuck into the maze

From deepest Gloucestershire (Cheltenham, in fact), Tudor Costigan conjures up a maze, and puts you in control of MAZE-MAN.



The program is easy to understand once you press RUN. You press 'a' to move up, 'z' to move down (the funny blob in line 320 of the instructions is a z), 'm' to move right and 'n' to move left. The reason for choosing three keys will become clear when you try the program out.

You get 100 points for an 'o' and 200 points for an explanation mark, with a miserable 10 for an asterisk.

Variables

0010-0070: Initialisation of variables and the like. POKE 23692,255 cures the Spectrum of its habit of asking 'scroll?' every time you blink. The subroutine at line 4300 draws the design shown at the beginning of the program.

0090-0185: These print the title and ask for the 'skill level', with suitable error messages. 'ww' is used as a delay loop later in the program.

0190-0210: These define the user graphics, using DATA statements from lines 3700 onwards. In the listing, the graphics are shown as the letter they are, rather than the little things you'll have flashing around your screen when it gets underway.

0230-0440: This section of the program provides instructions if they are needed.

0510-1630: These set up the position of the 'ghost' in 'l' and 'k', and then printout the board. Originally, when Tudor first wrote this, each 'character position' of the board had a different DATA element, but he found this too slow. Tudor says that the design of the board has been 'extensively tested' at his school, and there have been no complaints! You can make the elements of the lines different colours, puting the colours directly into the DATA statements, as described in your manual. The CHR$ 17; CHR$ 8 in line 1420 allows the old attributes of the paper to show through (see your manual, pages 111 and 114). Lines 1550 to 1610 print the title and scores in varying degrees of noticeability. Line 1620 prints the amount of man left in the middle of the board. Line 1630 sets the paper colour to green, the background colour of the insides of the maze.

Variables used:

n — high score
n$ — high score
j — code of object 'underneath' ghost
kk — number of lives left
i — to count objects eaten this 'frame'
b — score
d$ — stores previous key pressed
v — loop variable
c$ — general input variable
q,z — loop variables
d — loop variable
a$ — stores data for board printer
s — loop variable
x,y — position of maze man
k,l — position of ghost
ss, sd — contents of proposed position
q$ — score and colour control
aa,bb,cc,dd — length of notes
qa,qb — added to k and l produce old position

```
   1 REM MAZE MAN
   2 REM BY T.M.COSTIGAN
  10 LET n=47870: LET n$=STR$ n:
 POKE 23692,255
  20 LET j=32: LET kk=3: LET i=0
 LET b=0: LET d$=""
  30 BORDER 6: PAPER 7: INK 2: C
LS
  60 GO SUB 4300: FOR u=0 TO 5:
PRINT : NEXT u
  70 INK 0
```

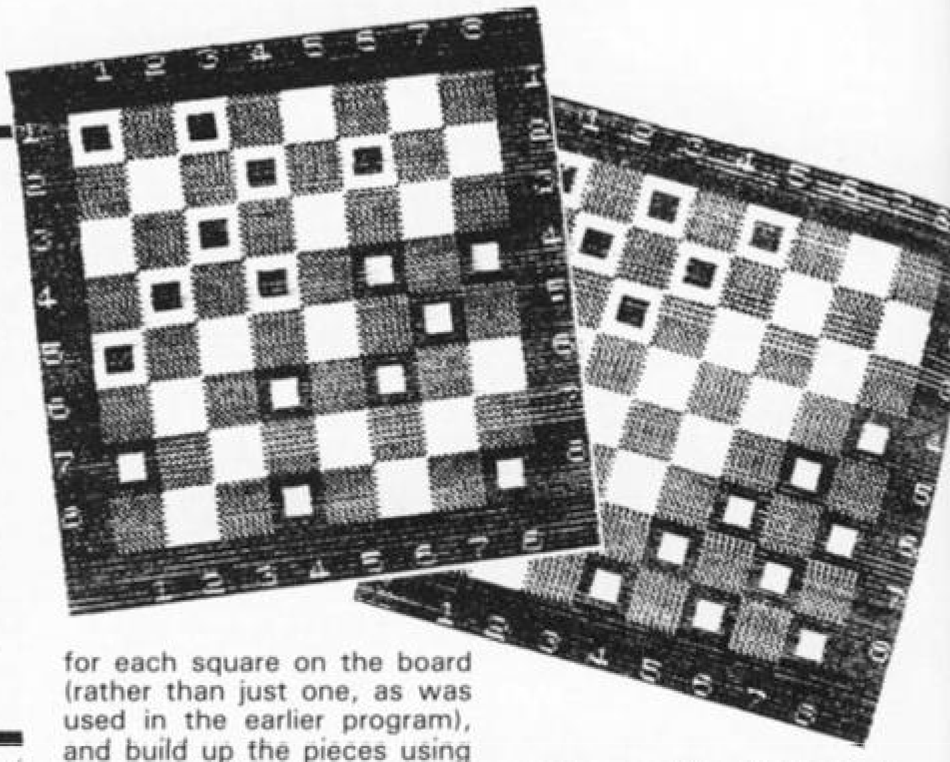
```
 90 PRINT FLASH 1; INK 1;"    MA
ZE MAN - By T.M.Costigan     "
 100 PRINT : PRINT
 110 PRINT "Skill Level?"
 120 PRINT "   Hard,Medium,Easy (
H,M,E) "
 130 PAUSE 0
 140 LET c$=INKEY$
 145 IF INKEY$<>"" THEN GO TO 14
5
 150 IF c$<>"h" AND c$<>"H" AND
c$<>"m" AND c$<>"M" AND c$<>"e"
AND c$<>"E" THEN PRINT AT 11,0;"
 H,M,or E only please!": GO TO 1
30
 160 IF c$="h" OR c$="H" THEN LE
T ww=0
 170 IF c$="m" OR c$="M" THEN LE
T ww=25
 180 IF c$="e" OR c$="E" THEN LE
T ww=50
 185 PRINT AT 10,27;c$
 190 RESTORE 3700
 200 FOR q=1 TO 6: READ x$
 205 FOR z=0 TO 7: READ x: POKE
USR x$+z,x: NEXT z
 210 NEXT q
 220 IF rc<>0 THEN GO TO 450
 230 PRINT : PRINT "Do you wish
instructions (Y/N)"
 240 IF INKEY$="" THEN GO TO 240
 250 LET c$=INKEY$
 260 IF c$="n" OR c$="N" THEN GO
TO 450
 270 IF c$<>"y" AND c$<>"Y" THEN
GO TO 240
 290 CLS
 300 PRINT : PRINT BRIGHT 1;"
   INSTRUCTIONS "
 310 PRINT : PRINT "To move up",
"press 'a'"
 320 PRINT "To move down","press
 'z'"
 330 PRINT "To move right","pres
s 'm'"
 340 PRINT "To move left","press
 'n'"
 360 PRINT : PRINT " * = 10 poin
ts"
 370 PRINT " O = 100 points"
 380 PRINT " ! = 200 points"
 430 PRINT : PRINT FLASH 1;"
PRESS ANY KEY TO CONTINUE     "
 435 IF INKEY$<>"" THEN GO TO 43
5
 440 IF INKEY$="" THEN GO TO 440
 510 CLS : LET l=6: LET k=19
 530 RESTORE
 650 DATA "#####################
#"
 660 DATA "#*******************
#"
 670 DATA "#*###*####*####*###*
#"
 680 DATA "#O###*************###O
#"
 690 DATA "#*****#*#######*#*****
#"
 700 DATA "#*###*#*********#*###*
#"
 710 DATA "#*****#####*#####*****
#"
 720 DATA "#####*#*#        #*####
#"

 730 DATA "     *#*#*#     #*#*#*
"
 740 DATA "#####*#*#*#     #*#*#*##
#"
 750 DATA "#*****#*########*#*****
#"
 760 DATA "#*#*#*****  *******#*#*
#"
 770 DATA "#O#*#*##*####*##*#*#O
#"
 780 DATA "#*#*#*************#*#*
#"
 790 DATA "#*******########******
#"
 800 DATA "#*####*########*####*
#"
 810 DATA "#*####*########*####*
#"
 820 DATA "#********************
#"
 830 DATA "#####################
#"
 1390 PAPER 7: CLS : PAPER 4: INK
 1
 1400 FOR d=1 TO 19
 1410 READ e$: FOR s=1 TO 22: IF
e$(s)="#" OR e$(s)=" " THEN PAPE
R 7: INK 4
 1412 IF d=12 AND s=11 THEN PAPER
 4
 1415 PRINT AT d,8+s;e$(s);: PAPE
R 4: INK 1
 1420 NEXT s: PRINT CHR$ 17,CHR$
5
 1460 NEXT d
 1490 PAPER 7: INK 2
 1550 PRINT AT 1,0; FLASH 1;"Maze
 man"; INVERSE 1;AT 2,3;"By"; IN
VERSE 0;AT 3,1;"T.M.C."
 1600 PRINT AT 5,0; INVERSE 1;"HI
 SCORE";AT 6,1; INVERSE 0; PAPER
 7;n$
 1610 PRINT AT 8,1; INVERSE 1;"SC
ORE";AT 9,1;STR$ b
 1620 PRINT AT 9,19;STR$ kk;"€"
 1630 PAPER 4
 1660 FOR s=0 TO 200: NEXT s
 1710 LET y=18: LET x=20: PRINT A
T y,x;"●"
 1750 FOR u=0 TO ww: NEXT u
 1755 BEEP .02,4
 1760 LET x$=INKEY$
 1765 IF x$="" THEN LET x$=INKEY$
 1770 IF x$<>"" THEN LET d$=""
 1771 PRINT AT y,x;" "
 1780 IF x$="m" OR x$="M" OR d$="
m" OR d$="M" THEN GO TO 1800
 1782 IF x$="n" OR x$="N" OR d$="
n" OR d$="N" THEN GO TO 1820
 1784 IF x$="a" OR x$="A" OR d$="
a" OR d$="A" THEN GO TO 1840
 1786 IF x$="z" OR x$="Z" OR d$="
z" OR d$="Z" THEN GO TO 1860
 1788 GO TO 2160
 1800 LET x=x+1
 1810 LET ss=CODE (SCREEN$ (y,x))
 : IF ss<>32 THEN GO TO 1900
 1812 PRINT AT y,x;"€"
 1815 GO TO 2100
 1820 LET x=x-1
 1830 LET ss=CODE (SCREEN$ (y,x))
 : IF ss<>32 THEN GO TO 1950
 1832 PRINT AT y,x;"Э"
 1835 GO TO 2100
 1840 LET y=y-1
 1850 LET ss=CODE (SCREEN$ (y,x))
 : IF ss<>32 THEN GO TO 2000
 1852 PRINT AT y,x;"Ѡ"
 1855 GO TO 2100
 1860 LET y=y+1
 1870 LET ss=CODE (SCREEN$ (y,x))
 : IF ss<>32 THEN GO TO 2050
 1872 PRINT AT y,x;"ⴖ"
 1875 GO TO 2100
 1900 IF ss=42 THEN LET i=i+1: LE
T b=b+10: GO TO 1812
 1905 IF ss=79 THEN LET b=b+100:
BEEP .5,5: BEEP .5,7: GO TO 1812
 1910 IF ss=33 THEN LET b=b+200:
BEEP 1,5: BEEP 1,7: GO TO 1812
 1915 IF ss=35 THEN LET x=x-1: GO
 TO 1812
 1920 GO TO 1812
```

```
1950 IF ss=42 THEN LET i=i+1: LE
T b=b+10: GO TO 1832
1955 IF ss=79 THEN LET b=b+100:
BEEP .5,5: BEEP .5,7: GO TO 1832
1960 IF ss=33 THEN LET b=b+200:
BEEP 1,5: BEEP 1,7: GO TO 1832
1965 IF ss=35 THEN LET x=x+1: GO
 TO 1832
1970 GO TO 1832
2000 IF ss=42 THEN LET i=i+1: LE
T b=b+10: GO TO 1852
2005 IF ss=79 THEN LET b=b+100:
BEEP .5,5: BEEP .5,7: GO TO 1852
2010 IF ss=33 THEN LET b=b+200:
BEEP 1,5: BEEP 1,7: GO TO 1852
2015 IF ss=35 THEN LET y=y+1: GO
 TO 1852
2020 GO TO 1852
2050 IF ss=42 THEN LET i=i+1: LE
T b=b+10: GO TO 1872
2055 IF ss=79 THEN LET b=b+100:
BEEP .5,5: BEEP .5,7: GO TO 1872
2060 IF ss=33 THEN LET b=b+200:
BEEP 1,5: BEEP 1,7: GO TO 1872
2065 IF ss=35 THEN LET y=y-1: GO
 TO 1872
2070 GO TO 1872
2100 IF x$<>"" THEN LET d$=x$
2103 IF y=9 AND x=31 THEN PRINT
AT y,x;" ": LET x=9: PRINT AT 9,
9;"€"
2106 IF y=9 AND x=8 THEN PRINT A
T y,x;" ": LET x=30: PRINT AT 9,
30;"■"
2110 IF i=50 OR i=100 OR i=150 T
HEN PRINT AT 12,19; FLASH 1; PAP
ER 7;"!": LET i=i+1
2120 IF (i=60 OR i=110 OR i=160)
 AND (y<>12 AND x<>19) AND (l<>1
2 AND k<>19) THEN PRINT AT 12,19
;" "
2130 LET q$=CHR$ 17+CHR$ 7+STR$
b+CHR$ 17+CHR$ 4: PRINT AT 9,1;
INK 0;q$
2135 IF b>n THEN PRINT AT 6,1; I
NK 0;q$
2140 IF i=186 THEN PRINT AT 8,15
; BRIGHT 1; FLASH 1; PAPER 7;"WE
LL DONE": LET I=0: GO TO 2142
2141 GO TO 2150
2142 LET aa=.125: LET bb=.25: LE
T cc=.375: LET dd=.5
2143 FOR z=1 TO 2
2145 BEEP cc,0: BEEP aa,-4: BEEP
 dd,-4: BEEP dd,0: BEEP cc,-4: B
EEP aa,-4: BEEP dd,-4: BEEP cc,0
: BEEP aa,0: BEEP dd,2: BEEP dd,
-1: BEEP bb,0: BEEP bb,4: BEEP 1
,0
2146 NEXT z
2148 BEEP cc,-1: BEEP aa,0: BEEP
 dd,2: BEEP dd,2: BEEP bb,4: BEE
P bb,0: BEEP dd,2: BEEP dd,2: BE
EP cc,-1: BEEP aa,0: BEEP dd,2:
BEEP dd,2: BEEP bb,4: BEEP bb,0:
BEEP dd,2: BEEP dd,2: BEEP cc,0
: BEEP aa,-1: BEEP dd,-4: BEEP d
d,2: BEEP dd,0: BEEP dd,2: BEEP
dd,4: BEEP cc,0: BEEP aa,-1: BEE
P dd,-4: BEEP bb,5: BEEP bb,2: B
EEP dd,0: BEEP dd,-1: BEEP dd,0
2149 GO TO 510
2150 REM MOVING ALIEN
2160 PRINT AT y,x;"■"
2161 BEEP .02,0
2165 IF l-y>0 THEN GO TO 2410
2170 IF l-y<0 THEN GO TO 2510
2180 IF k-x>0 THEN GO TO 2210
2190 IF k-x<0 THEN GO TO 2310
2200 GO TO 2650
2210 LET k=k-1
2220 LET qa=1: LET qb=0
2240 LET sd=CODE (SCREEN$ (l,k))
2250 IF sd=35 THEN LET k=k+1: GO
 TO 2190
2260 GO TO 2610
2310 LET k=k+1
2320 LET qa=-1: LET qb=0
2340 LET sd=CODE (SCREEN$ (l,k))
2350 IF sd=35 THEN LET k=k-1: GO
 TO 1750
2360 GO TO 2610
2410 LET l=l-1
2420 LET qa=0: LET qb=1
2440 LET sd=CODE (SCREEN$ (l,k))
2450 IF sd=35 THEN LET l=l+1: GO
 TO 2170
2460 GO TO 2610
2510 LET l=l+1
2520 LET qa=0: LET qb=-1
2540 LET sd=CODE (SCREEN$ (l,k))
2550 IF sd=35 THEN LET l=l-1: GO
 TO 2180
2610 IF sd=32 THEN GO SUB 2700:
LET j=32: GO TO 1750
2620 IF sd=42 THEN GO SUB 2700:
LET j=42: GO TO 1750
2630 IF sd=79 THEN GO SUB 2700:
LET j=79: GO TO 1750
2640 IF sd=33 THEN GO SUB 2700:
LET j=32: GO TO 1750
2650 IF l=y AND k=x THEN GO SUB
2700: GO TO 2710
2660 GO TO 1750
2700 PRINT AT l+qb,k+qa;CHR$ j:
PRINT AT l,k;"■": RETURN
2710 LET kk=kk-1
2720 LET i=0
2730 FOR a=0 TO 7
2740 BEEP .02,0: BORDER a: BEEP
.02,1: PAPER 7-a: BEEP .02,4: CL
S
2750 NEXT a
2760 BORDER 6: BEEP .05,0: PAPER
 7: BEEP .05,1: INK 2: BEEP 2,3:
CLS
2770 IF kk>0 THEN GO TO 510
2780 PRINT "You ended with ";b;"
points."
2790 IF n<b THEN LET n=b: PRINT
"This is the new high score"
2800 PRINT ' FLASH 1; BRIGHT 1;"
         ANOTHER GAME?
"
2810 LET e$=INKEY$
2820 IF e$<>"y" AND e$<>"Y" AND
e$<>"n" AND e$<>"N" THEN GO TO 2
810
2830 LET rc=1: IF e$="y" OR e$="
Y" THEN GO TO 20
2840 STOP
3700 DATA "a",0,60,126,14,14,126
,60,0
3710 DATA "e",0,60,126,126,126,1
26,60,0
3720 DATA "f",0,60,126,126,126,8
4,84,0
3730 DATA "b",0,60,126,112,112,1
26,60,0
3740 DATA "c",0,36,102,102,126,1
26,60,0
3750 DATA "d",0,60,126,126,102,1
02,36,0
4300 FOR x=0 TO 255 STEP 5
4310 PLOT 127,87: DRAW x-127,88
4320 NEXT x
4330 FOR y=175 TO 0 STEP -5
4340 PLOT 127,87: DRAW 128,y-87
4350 NEXT y
4360 FOR x=255 TO 0 STEP -5
4370 PLOT 127,87: DRAW x-127,-87
4380 NEXT x
4390 FOR y=0 TO 175 STEP 5
4400 PLOT 127,87: DRAW -127,y-87
4410 NEXT y
4420 RETURN
```

Board games for your computer

In the last issue of ZX Computing, editor Tim Hartnell explained one way of writing board games for computer use. Here, he takes the topic a little further, and gives you two board games with full screen displays for the 16K ZX81.



Chess Board

The numbered chess-board shown here contains the heart of computer board games.

You can indicate any square on the board by referring to the number along the left hand side (such as 4) then the number along the top (such as 5). In this case, the lines numbered 4 (along the left hand side) and the line numbered 5 (along the top) meet at the square numbered 45. If you wish to move a piece, you can do so by entering the number of the square you're moving from, (such as 45), then the number you are moving to, (such as 54), and the computer can understand exactly what you are doing.

This was explained in my article in the October/November issue of ZX Computing. I went on to explain that the second great advantage of using a numbered board of this type was that when you move — in any direction — no matter where you are on the board, the *difference* in value of the 'from' square and that of the 'to' square is the same.

The predictable nature of the numerical difference between squares makes it fairly easy for the computer to work out its next moves. If you entered the 'Corner Checkers' game given in the last issue, you'll find it relatively easy to adapt it to become both of the board games given here, which have full screen displays.

The way to get a big display is to use four character cells for each square on the board (rather than just one, as was used in the earlier program), and build up the pieces using the 'chunky' graphic options on the ZX81.

When you run the programs, you'll see that PRINT AT is used to change just the squares moved to and from (and, when needed, to remove a captured piece), rather than have the board reprinted every time. This speeds the game up considerably.

Corner Checkers

Corner checkers follows the rules of draughts, except that you play by starting in the corners of the board, rather than at the ends, there are no multiple jumps, and no kings. Any piece may move in any diagonal direction. Captures are in draughts, by jumping over an opponent's piece into an empty square, always moving on the diagonal. As you can see from the sample run which follows the listing, the pieces look particularly effective, and you'll have no trouble following the game when you get the program underway.

Spanish Checkers

Once you've entered Corner Checkers, and saved it on tape, you can alter it to become the game Spanish Checkers, which is closer to 'real' draughts. In this, you move from left to right across the board; the computer from right to left. You play as in draughts, except that you can move in any diagonal direction (as though you had a board of kings), there are no kings, and no multiple jumps.

```
  10 REM CORNER CHECKERS
  20 GOSUB 9000
  30 GOSUB 8000
  35 SLOW
  40 GOSUB 7000
  50 GOSUB 6000
  60 IF HUM=7 THEN PRINT AT 19,0
;"YOU WIN";W
  70 IF COMP=7 THEN PRINT AT 19,
0;"I WIN";W
  80 GOTO 40
6000 REM **COMPUTER MOVE**
6010 FOR Z=88 TO 11 STEP -1
6020 IF A(Z)=C THEN GOTO 6050
6030 NEXT Z
6040 GOTO 6200
6050 LET Y=-11
6055 IF Z+Y>88 OR Z+Y<11 OR Z+2*
Y>88 OR Z+2*Y<11 THEN GOTO 6070
6060 IF A(Z+Y)=H AND A(Z+2*Y)=E
THEN GOTO 6100
6070 LET Y=-9*(Y=-11)+9*(Y=-9)+1
0*(Y=9)+(Y=100)
6080 IF Y<>0 THEN GOTO 6055
6085 NEXT Z
6090 GOTO 6200
6100 LET A(Z)=E
6110 LET A(Z+Y)=E
6120 LET A(Z+2*Y)=C
6130 LET COMP=COMP+1
6140 PRINT AT 2,22;"COMP.:  ";COM
P
6150 LET F=INT ((Z+Y)/10)
6160 LET G=Z+Y-10*F
6170 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6172 LET F=INT ((Z+2*Y)/10)
6174 LET G=Z+2*Y-10*F
6176 PRINT AT 2*G,2*F;"▛▜";AT 2*
G+1,2*F;"▙▟"
6180 LET F=INT (Z/10)
6182 LET G=Z-10*F
6184 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6190 RETURN
6200 REM **NON-CAPTURE MOVE**
6210 FOR Z=1 TO 200
6220 LET K=INT (RND*78)+11
```

```
6230 IF A(K)=C THEN GOTO 6260
6240 NEXT Z
6250 GOTO 6500
6260 LET Y=-11
6280 IF A(K+Y)=E THEN GOTO 6330
6290 LET Y=-9*(Y=11)+9*(Y=-9)+11
*(Y=9)+(Y=100)
6300 IF Y<>0 THEN GOTO 6270
6310 NEXT Z
6320 GOTO 6500
6330 IF K+2*Y>88 OR K+2*Y<11 THE
N GOTO 6400
6340 IF A(K+2*Y)=H THEN GOTO 624
0
6350 IF K-2*Y<11 OR K-2*Y>88 THE
N GOTO 6400
6360 IF A(K-2*Y)=H THEN GOTO 624
0
6400 LET A(K+Y)=C
6410 LET A(K)=E
6420 LET F=INT ((K+Y)/10)
6430 LET G=K+Y-10*F
6440 PRINT AT 2*G,2*F;"▛▜";AT 2*
G+1,2*F;"▙▟"
6450 LET F=INT (K/10)
6460 LET G=K-10*F
6470 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6480 RETURN
6500 FOR G=1 TO 200
6510 LET K=INT (RND*78)+11
6520 IF A(K)=C THEN GOTO 6600
6530 NEXT G
6540 PRINT AT 0,0;"I CONCEDE THE
 GAME"
6550 STOP
6600 IF A(K-11)=E THEN LET Y=-11
6610 IF A(K-11)=E THEN GOTO 6400
6620 IF A(K-9)=E THEN LET Y=-9
6630 IF A(K-9)=E THEN GOTO 6400
6640 GOTO 6540
7000 REM **PLAYER MOVE**
7010 PRINT AT 20,0;"ENTER YOUR M
OVE AS ""3344"""
7020 INPUT A$
7030 IF LEN A$<>4 THEN GOTO 7020
7040 PRINT AT 20,0;"
"
7050 LET A1=VAL A$(1)
7055 LET A2=VAL A$(2)
7060▯LET B1=VAL A$(3)
7065 LET B2=VAL A$(4)
7070 LET A(10*B1+B2)=H
7080 LET A(10*A1+A2)=E
7090 PRINT AT 2*B2,2*B1;"▄▄";AT
2*B2+1,2*B1;"▀▀"
7100 PRINT AT 2*A2,2*A1;"  ";AT
2*A2+1,2*A1;"  "
7110 IF ABS (A1-B1)=1 THEN RETUR
N
7120 LET HUM=HUM+1
7130 PRINT AT (A2+B2),(A1+B1);"
";AT A2+B2+1,A1+B1;" "
7140 PRINT AT 0,22;"HUMAN: ";HUM
7990 RETURN
8000 REM PRINT BOARD - START
8040 FOR Z=8 TO 1 STEP -1
8060 FOR X=1 TO 8
8070 IF A(10*Z+X)=H THEN PRINT A
T 2*X,2*Z;"▄▄";AT 2*X+1,2*Z;"▀▀"
8080 IF A(10*Z+X)=C THEN PRINT A
T 2*X,2*Z;"▛▜";AT 2*X+1,2*Z;"▙▟"
8090 IF A(10*Z+X)=B THEN PRINT A
T 2*X,2*Z;"▒▒";AT 2*X+1,2*Z;"▒▒"
8110 NEXT X
8120 NEXT Z
8130 RETURN
8990 STOP
8999 STOP
9000 FAST
9005 DIM A(100)
9010 LET H$="1113152224313334251"
```

```
9020 LET C$="8886776888475665748"
9030 LET B$="12141618212325273233
4363841434547525456586163656 7727
47678816383587"
9040 LET E$="6273645546372817263
544536271"
9050 FOR Z=1 TO 100
9060 LET A(Z)=9
9070 NEXT Z
9080 LET H=CODE "H"
9090 LET C=CODE "C"
9095 LET B=CODE "▒"
9100 LET E=CODE " "
9105 FOR Z=1 TO 9
9110 LET A(VAL H$( TO 2))=H
9120 LET A(VAL C$( TO 2))=C
9130 LET H$=H$(3 TO )
9135 LET C$=C$(3 TO )
9140 NEXT Z
9150 FOR Z=1 TO 32
9160 LET A(VAL B$( TO 2))=B
9170 LET B$=B$(3 TO )
9180 NEXT Z
9190 FOR Z=1 TO 14
9200▯LET A(VAL E$( TO 2))=E
9210 LET E$=E$(3 TO )
9220 NEXT Z
9230 LET COMP=0
9240 LET HUM=0
9400 PRINT AT 0,0;"   1 2 3 4 5
 6 7 8  ";AT 1,1;"
 ";AT 18,0;"
 ";AT 19,0;"   1 2 3 4 5 6 7 8
"
9410 FOR Z=1 TO 17
9420 PRINT AT Z,1;"▒";AT Z,18;"▒
"
9430 IF 2*INT (Z/2)=Z THEN PRINT
 AT Z,0;CHR$ (156+Z/2);AT Z,19;C
HR$ (156+Z/2)
9440 IF 2*INT (Z/2)<>Z THEN PRIN
T AT Z,0;"▒";AT Z,19;"▒"
9450 NEXT Z
9500 RETURN
```

```
  10 REM SPANISH CHECKERS
  20 GOSUB 9000
  30 GOSUB 8000
  35 SLOW
  40 GOSUB 7000
  50 GOSUB 6000
  60 IF HUM=7 THEN PRINT AT 19,0
;"YOU WIN";W
  70 IF COMP=7 THEN PRINT AT 19,
0;"I WIN";W
  80 GOTO 40
6000 REM **COMPUTER MOVE**
6010 FOR Z=88 TO 11 STEP -1
6020 IF A(Z)=C THEN GOTO 6050
6030 NEXT Z
6040 GOTO 6200
6050 LET Y=-11
6055 IF Z+Y>88 OR Z+Y<11 OR Z+2*
Y>88 OR Z+2*Y<11 THEN GOTO 6070
6060 IF A(Z+Y)=H AND A(Z+2*Y)=E
THEN GOTO 6100
6070▯LET Y=-9*(Y=-11)+9*(Y=-9)+1
0*(Y=9)+(Y=100)
6080 IF Y<>0 THEN GOTO 6055
6085 NEXT Z
6090 GOTO 6200
6100 LET A(Z)=E
6110 LET A(Z+Y)=E
6120 LET A(Z+2*Y)=C
6130 LET COMP=COMP+1
6140 PRINT AT 2,22;"COMP.: ";COM
P
6150 LET F=INT ((Z+Y)/10)
6160 LET G=Z+Y-10*F
6170 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6172 LET F=INT ((Z+2*Y)/10)
6174 LET G=Z+2*Y-10*F
```

```
6176 PRINT AT 2*G,2*F;"▚▄▞";AT 2*
G+1,2*F;"▞▀▚"
6180 LET F=INT (Z/10)
6182 LET G=Z-10*F
6184 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6190 RETURN
6200 REM **NON-CAPTURE MOVE**
6210 FOR Z=1 TO 200
6220 LET K=INT (RND*78)+11
6230 IF A(K)=C THEN GOTO 6260
6240 NEXT Z
6250 GOTO 6500
6260 LET Y=-11
6280 IF A(K+Y)=E THEN GOTO 6330
6290 LET Y=-9*(Y=11)+9*(Y=-9)+11
*(Y=9)+(Y=100)
6300 IF Y<>0 THEN GOTO 6270
6310 NEXT Z
6320 GOTO 6500
6330 IF K+2*Y>88 OR K+2*Y<11 THE
N GOTO 6400
6340 IF A(K+2*Y)=H THEN GOTO 624
0

6350 IF K-2*Y<11 OR K-2*Y>88 THE
N GOTO 6400
6360 IF A(K-2*Y)=H THEN GOTO 624
0
6400 LET A(K+Y)=C
6410 LET A(K)=E
6420 LET F=INT ((K+Y)/10)
6430 LET G=K+Y-10*F
6440 PRINT AT 2*G,2*F;"▚▄▞";AT 2*
G+1,2*F;"▞▀▚"
6450 LET F=INT (K/10)
6460 LET G=K-10*F
6470 PRINT AT 2*G,2*F;"  ";AT 2*
G+1,2*F;"  "
6480 RETURN
6500 FOR G=1 TO 200
6510 LET K=INT (RND*78)+11
6520 IF A(K)=C THEN GOTO 6600
6530 NEXT G
6540 PRINT AT 0,0;"I CONCEDE THE
 GAME"
6550 STOP
6600 IF A(K-11)=E THEN LET Y=-11
6610 IF A(K-11)=E THEN GOTO 6400
6620 IF A(K-9)=E THEN LET Y=-9
6630 IF A(K-9)=E THEN GOTO 6400
6640 GOTO 6540
7000 REM **PLAYER MOVE**
7010 PRINT AT 20,0;"ENTER YOUR M
OVE AS ""3344"""
7020 INPUT A$
7030 IF LEN A$<>4 THEN GOTO 7020
7040 PRINT AT 20,0;"
"
7050 LET A1=VAL A$(1)
7055 LET A2=VAL A$(2)
7060 LET B1=VAL A$(3)
7065 LET B2=VAL A$(4)
7070 LET A(10*B1+B2)=H
7080 LET A(10*A1+A2)=E
7090 PRINT AT 2*B2,2*B1;"▄█▄";AT
2*B2+1,2*B1;"▀█▀"
7100 PRINT AT 2*A2,2*A1;"  ";AT
2*A2+1,2*A1;"  "
7110 IF ABS (A1-B1)=1 THEN RETUR
N
7120 LET HUM=HUM+1
7130 PRINT AT (A2+B2),(A1+B1);"
 ";AT A2+B2+1,A1+B1;"  "
7140 PRINT AT 0,22;"HUMAN: ";HUM
7990 RETURN
8000 REM PRINT BOARD - START
8040 FOR Z=8 TO 1 STEP -1
8060 FOR X=1 TO 8
8070 IF A(10*Z+X)=H THEN PRINT A
T 2*X,2*Z;"▄█▄";AT 2*X+1,2*Z;"▀█▀"
8080 IF A(10*Z+X)=C THEN PRINT A
T 2*X,2*7;"▚▄▞";AT 2*X+1,2*Z;"▞▀▚"
8090 IF A(10*Z+X)=B THEN PRINT A
T 2*X,2*Z;"▒▒";AT 2*X+1,2*Z;"▒▒"
8110 NEXT X
8120 NEXT Z
8130 RETURN
8998 STOP
8999 STOP
9005 DIM A(100)
9010 LET H$="111315172224262831З
33537"
9020 LET C$="828486887173757762б
46668"
9030 LET B$="1214161821232527323
4363841434547525456586163656577727
4767881838587"
9040 LET E$="5153555742444648"
9050 FOR Z=1 TO 100
9060 LET A(Z)=9
9070 NEXT Z
9080 LET H=CODE "H"
9090 LET C=CODE "C"
9095 LET B=CODE "█"
9100 LET E=CODE " "
9105 FOR Z=1 TO 12
9110 LET A(VAL H$( TO 2))=H
9120 LET A(VAL C$( TO 2))=C
9130 LET H$=H$(3 TO )
9135 LET C$=C$(3 TO )
9140 NEXT Z
9150 FOR Z=1 TO 32
9160 LET A(VAL B$( TO 2))=B
9170 LET B$=B$(3 TO )
9180 NEXT Z
9190 FOR Z=1 TO 8
9200 LET A(VAL E$( TO 2))=E
9210 LET E$=E$(3 TO )
9220 NEXT Z
9230 LET COMP=0
9240 LET HUM=0
9400 PRINT AT 0,0;"  1 2 3 4 5 
6 7 8  ";AT 1,1;"                  
 ";AT 18,0;"                    
 ";AT 19,0;"  1 2 3 4 5 6 7 8  
"
9410 FOR Z=1 TO 17
9420 PRINT AT Z,1;"█";AT Z,18;"█
"
9430 IF 2*INT (Z/2)=Z THEN PRINT
 AT Z,0;CHR$ (156+Z/2);AT Z,19;C
HR$ (156+Z/2)
9440 IF 2*INT (Z/2)<>Z THEN PRIN
T AT Z,0;"█";AT Z,19;"█"
9450 NEXT Z
9500 RETURN
```

	1	2	3	4	5	6	7	8
8	81	82	83	84	85	86	87	88
7	71	72	73	74	75	76	77	78
6	61	62	63	64	65	66	67	68
5	51	52	53	54	55	56	57	58
4	41	42	43	44	45	46	47	48
3	31	32	33	34	35	36	37	
2	21	22	23	24	25	26	27	28
1	11	12	13	14	15	16	17	18

Under the ZX-scope

Reviewer Nick Pearce takes a close look at a number of ZX81 programs, including Artic's Toolkit, a word processor program from Tasman Software and Richard Shepherd's 'Super Space Mission'.

Toolkit — Artic Computing

For the basic programmer, Toolkit from Artic provides a total of nine functions to ease the labour of program writing, including the usual renumbering facility.

After loading, Toolkit automatically lowers RAMTOP and sets itself up in the last 2¼K of memory — a definite advantage on some other toolkits which require the user to reset RAMTOP before loading, something which is easily forgotten.

Toolkit's facilities are obtained through USR calls. Prompts are given on the screen and the program is fairly straightforward to use, special error codes being given to identify the problem if anything goes wrong.

The instructions that come with the cassette are closely typed on one side of an A5 sheet of paper, and although the facilities of the program are adequately explained, the presentation could be better. Some of the error codes, for example, are difficult to find, and a listing of the codes and their meaning would be helpful.

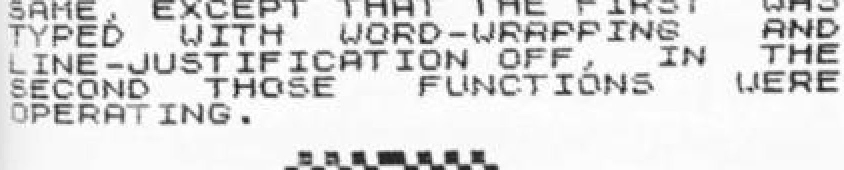
Renumbering

The renumber routine is most impressive. It renumbers a program, including GOTOs, GOSUBs, RUNs and LISTs from any start line and in any step size — the screen prompts the necessary commands. After renumbering any computed GOTOs or GOSUBs (eg GOTO 100 x LEN A$) are displayed so that these can be numbered manually by the user. The Toolkit will cope with jumps to non-existent line numbers by remembering the jump command with the number of the line next after the non-existent line. Renumbering is carried out so quickly as to be almost instantaneous on even long basic programs.

Apart from renumbering, the Toolkit includes the facilities to delete groups of lines in the program (prompting for the first and last of the line numbers to be deleted); to display the amount of memory available in bytes; to list the current values of all string and numerical variables (except arrays and loop control variables); and find any string up to 255 characters in length and list every line in the program containing that string. The replace command allows any string to be replaced by any other string, which need not necessarily be the same length, eg PRINT can be replaced by LPRINT. By saving the current program below RAMTOP, another progam can be loaded, allowing two programs to be joined together. Another function is REMKILL which removes all REM statements from a program — useful if you are running short of memory, or want to make your programs quicker to LOAD and SAVE.

This must be one of the better toolkits around for the ZX81. It is easy to use, executes commands quickly and accurately, and all of its nine functions are likely to have some value both for the novice and for the more experienced basic programmer. Toolkit takes up only just over 2K of memory, quite impressive considering the range of functions available.

Toolkit costs £5.95p and is available from Artic Computing Ltd., 396 James Reckitt Avenue, Hull, North Humberside.



TASWORD

ZX81 WORD PROCESSOR

THESE TWO PARAGRAPHS ARE THE SAME, EXCEPT THAT THE FIRST WAS TYPED WITH WORD-WRAPPING AND LINE-JUSTIFICATION OFF, IN THE SECOND THOSE FUNCTIONS WERE OPERATING.

THESE TWO PARAGRAPHS ARE THE SAME, EXCEPT THAT THE FIRST WAS TYPED WITH WORD-WRAPPING AND LINE-JUSTIFICATION OFF, IN THE SECOND THOSE FUNCTIONS WERE OPERATING.

TASWORD— Tasman Software.

Tasword from Tasman Software is a program that makes a credible word processor out of your ZX81. The program demonstrates admirably the capabilities of a word processing system, although the inherent limitations of the ZX81 do pose restrictions — there are no lower-case characters for example. Up to 320 lines of text can be handled in the Tasword text file and 22 lines of the file are displayed at any one time on the television screen.

Shift Keys

The character keys of the ZX81 keyboard are used in the normal way to write onto the file. Control of Tasword's word processing operations are achieved through the "shift" keys. Shift (or control) keys are used to delete or insert lines or characters, obtain the graphics characters of the ZX81, scroll the text, move complete lines, and perform most of the other usual word processing operations. Unless overridden by a control key, Tasword "Word wraps" automatically, i.e. if the last word overflows the end of a line, then the whole word is transferred to the next line, and the finished line is spread out across the screen (justified) to give the line a neat appearance.

The program is well thought out and easy to master, helped by the logical assignment of control keys to word processor functions, for example "< >" to centralise headings on the screen, "AND" to insert new lines, words or characters onto the file, and the ARROW keys 5, 6, 7 and 8 to move the cursor around the screen.

Saving the program

The STOP function stops Tasword running and permits a number of options to be selected. The program can be saved on cassette, either with or without the text file, the text file can be cleared, or control passed to the ZX81 ROM. This is a particularly useful innovation since it permits other tasks to be carried out while still retaining Tasword and the text file in memory, eg calculations can if necessary be carried out without losing the program or text.

Instructions for Tasword come in the form of a small booklet which is clear and well written and, inevitably of course, produced using Tasword itself. Somehow Tasman have produced a printout for the booklet which is far superior to anything I have been able to manage on the Sinclair printer!

On the reverse side of the cassette is Tasword Tutor, a well presented teaching aid to the Tasword functions. A nice touch to complete this package, although largely unnecessary as most users will find Tasword easy to master from the instruction booklet alone.

I was a little disappointed that Tasword lacks the facility to move blocks of text around

in the file — I inevitably have to rearrange the sequence of paragraphs whilst drafting articles or letters.

The overall usefulness of this program is limited by the ZX81 itself, particularly the lack of lower case characters and restricted text line length (32 characters), and the Sinclair printer. However, for the price it is an impressive package. If you want to use your ZX81 for serious word processing there are software packages available that offer lower case characters, although they are rather slow in operation; otherwise you will have to dig very much deeper into your pocket for suitable hardware — perhaps including an RS232 interface for a "proper" printer.

I had no LOADing problems with any of the software cassettes in this review except Tasword, which refused to load until I had made significant adjustments to the head alignment of my cassette recorder, indicative perhaps of head misalignment on Tasman's recording equipment. Tasman will change cassettes that do not load, and ask that you specify the make and model of your cassette recorder if a replacement is required.

Tasword costs £6.50p and is available from Tasman Software, 17, Hartley Crescent, Leeds.

Super Space Mission — Richard Shepherd Software

And now to some new games for the ZX81.

Super Space Mission is an arcade style game, but with plenty of added interest, including some impressive introductory graphics.

You start on the launch pad and initiate countdown, after which take-off is automatic. Control of your space craft is handed back to you in deep space. After testing the controls you are on your own and may proceed with the mission, which is to destroy an alien fleet. As well as the aliens, asteroids are a regular threat, and your lasers can be damaged as you fight them off, reducing your firing capability.

Pilot rating

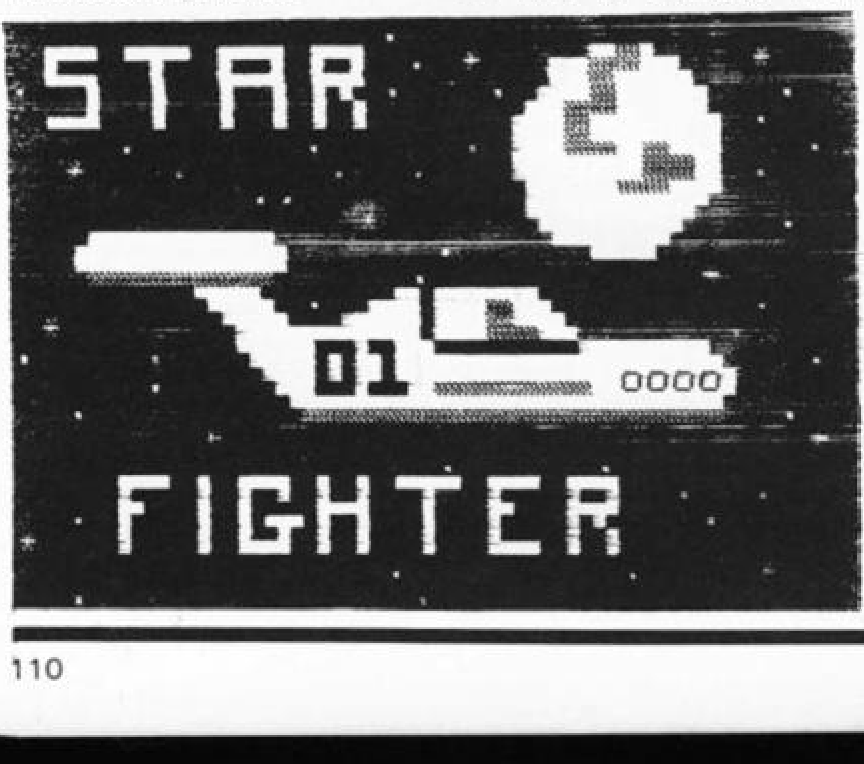
Fast reflexes and luck are essential if you are to stand any chance of destroying the alien ships as they swoop around you. With yours truly at the controls the mission usually had to be aborted uncompleted, or ended after suffering fatal damage. A final mission report includes pilot rating, mine usually read "lousy shot", although I did get a "not bad" after one brilliantly executed mission, and you are given a scwre (sic).

If your think the game is getting too easy, a more difficult of the seven skill levels can be selected up to the hardest — suicidal.

There is at least one bug in the program — apart from the spelling mistake in the mission report. The game occasionally "jammed" during landing, in which case it had to be reloaded; and a mission would sometimes come to an end before it should.

A fast moving arcade style game which goes a bit further than the usual invaders program, with some nice touches and particularly good graphics. Rather expensive, however, and the bugs do need to be sorted out.

Super Space Mission costs £6 and is available from Richard Shepherd Software, 22 Green Leys, Maidenhead, Berkshire.



Byte Man — Mindseye

Impressive graphics are a feature of this cassette too, and priced at only £2.95 for three games it is good value for money. Another good feature is the game instructions; these are not written down, but the player can opt for comprehensive and well presented on-screen instructions at the start of each game.

Byte Man is a version of Pac-man, the video game that has taken America by storm and is now rapidly gaining popularity here. Two other ZX81 versions are reviewed later. In Byte Man, you use the arrow keys to move around the maze, eating up dots, or food pills, worth five points each as you go. The maze monsters, Byteman and his two brothers, will eat you if they catch you, and you have three lives. If you can get to one of the four "stars" in the maze you can take your revenge and chase the monsters for a short while, adding to your score if you catch them.

Monster's lair

You start near the centre of the maze, and very close to the monster's lair. I found it extremely difficult to get more than a few moves away from the start without being jumped on by one of them. Once out of the vicinity of the start, however, the game becomes easier. Your final score can be entered into an on-screen league table so that several people may have a competition.

Star fighter

The second program on the Mindseye cassette is a "Defenders" type game in which you are the pilot of a small scout ship patrolling space. Waves of Sylvian fighters try to reach Earth, and your job is to destroy as many as you can. Points are gained for each enemy ship and each missile shot down. The Sylvian fighter attacks become faster and more frequent, and if things get too hot you can "warp out" into a safe (for a short while) area, but only once per life — you have three lives per game. Because you lose points for the fighters you let through, you can play for a long time and end with a very low final score — zero on my first game (you've probably already guessed that I'm not particularly good at this sort of thing), but I did manage 90 after some practice.

Again, good graphics, your scout ship explodes convincingly when hit, and a score table is kept. An addictive game requiring skill and nimble fingers to achieve a high score.

Bomber

Finally, Bomber is a simple game in which you have 30 bombs to destroy a dam. You fly repeatedly over the dam at the same altitude, and only one bomb can be in the air at any time.

Some parts of the dam are worth more than others, and a certain amount of skill is required to achieve a high score. However, not a game to get the adrenalin flowing; your plane is never in danger and it would be more exciting if, for example, you were shot at by gun emplacements near the dam. Your score is rated. I thought I had done well to receive "High score" for my first go. A few games later I concluded that it is impossible *not* to get this rating and I think Mindseye would be well advised to review the rating system.

Three games on one cassette makes this software good value for money. The use of graphics is particularly good, each game is preceeded by a relevant screen display — a picture of your aircraft in Bomber for example.

This cassette will give the newcomer a good idea of the interactive arcade type programs currently available for the ZX81.

Byte Man costs £2.95 and is available from Mindseye, 12 North Grove Drive, Leeds.

Zedman + Spacers — Babtech.

Zedman is another Pac-man type game. The maze is smaller and less complicated in this version. You are a mouse and move around the maze eating dots, again using the arrow keys (5, 6, 7 and 8). Also moving around the maze are ghosts (stars) that can eat you, but if you eat a magic pill (a dollar — there are four, one in each corner of the maze) then the ghosts become edible for a short while.

Slippery creatures

You select a skill rating from one to ten, ten being very fast, and an added complication is that when edible the ghosts are slippery creatures and if you rush at them they will slip past you. A fast moving, absorbing and addictive game.

Babtech have produced a good version of Pac-man for the ZX81, but I was not over impressed with Spacers, the second game on this cassette.

Spacers is an invaders type program in which you use your laser gun to shoot down the descending alien hordes. There are some nice touches — you can select missile speed from the range 1 to 10, and missile density from 1 to 4, and you can also choose to renew your shields for each attack wave. However, the game is rather spoilt by the slow, jerky response of your laser base to the movement keys (keys 5 and 8).

I liked Zedman, but have seen better invader games than Spacers.

Zedman is available from Babtech, 3 Baberton Mains View, Edinburgh EH14 3BR.

Gobbleman, Namtir Raiders — Artic Computing

In Gobbleman, Artic's version of Pac-man, you are doomed to roam (spelt rkam, the second spelling error in the cassettes reviewed for this issue — so much for computers in education. . .) through the haunted maze eating food dots until you are eaten by a ghost. Again, if you eat a magic star you can take your revenge on the ghosts for a short while.

I did not think this game was up to Artic's usual very high standard. Movement is slow and jerky, and the speed at which you can move through the maze is not variable. I like the keyboard scanning arrangement Artic have used — any of the top keys move you upwards in the maze, the bottom row move you down, and the centre two rows are split left and right. Much easier than using specific keys for movement on the touch sensitive keyboard of the Sinclair.

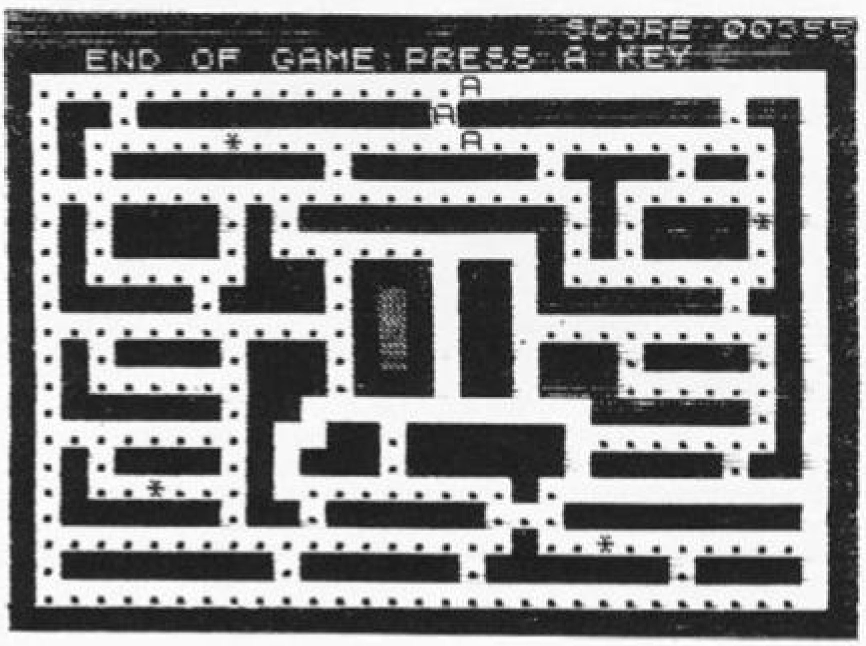
Raiders

The other Artic cassette, and the last game in this issue's ZX81 software review is Namtir Raiders (author J. Ritman — get it?), a superb invaders type program. In this one you do not hide behind bases or shields on the ground, but can get up amongst the descending armies of invaders to shoot them down.

There are four invader fleets, ships of the first fleet are worth 100 points each, whilst ships of the fourth are worth 1000 points. You can select one of three speeds and have four bases to start (lives) with an extra base to be won if you reach the fourth fleet. A keyboard scanning arrangement similar to that employed in Gobbleman is used, so you do not have to hunt for specific keys.

Tactics

You need fast reflexes to blast at the invaders and at the same time evade their bomb attacks. Tactics are important for a high score — if you destroy the first fleet you move on to the next and so on up to the fourth, but if one or two of a fleet escape, that same fleet regenerates again . . . and again . . . and again . . .



It is easier to knock out ships of the first and second fleets, the third I found practically impossible; and as you wipe out more ships the ramainder get faster and faster.

A first rate invaders type program. If you like the arcade game, stop pouring silver into those machines, plug in your ZX81 and get blasting away at the Namtir Raiders.

Gobbleman and Namtir Raiders cost £3.95 each, and are obtainable from Artic Computing Ltd., 396 James Reckitt Avenue, Hull, North Humberside.

RETA for retailers — Michael Cox Information Services

Turning from games to a serious application for the ZX81, Reta is a decision modelling program for small businesses. It is an ambitious program designed to help retailers estimate the effect of price changes and advertising expenditure on business profitability. The author also suggests that Reta will be a valuable teaching aid to highlight some of the problems faced by retailers.

Information

It is, I suppose, basically a VISICALC type program which has been formulated to manipulate the cost and sales information of a business. You are first asked to enter some basic information about your business. This forms the base model from which the effects of changes can be estimated. For most businesses this information will be mainly factual; product prices, quantities and costs, although some will require estimation. There then follows routines that allow the model to be recalculated in respect of price changes and price response (the predicted effect of a price change), and advertising expenditure and response (an estimate of the effect of advertising).

The program demonstrates the effect of price and/or advertising changes in terms of a percentage change on the base model. You can then move on to look at the model results in detail. This is the core of the Reta system. It is menu driven and is used to study the effect on profits of the price or advertising changes you have considered, enabling you to analyse *why* these effects have occurred.

The program itself is long, over 13K, and takes over 6 minutes to LOAD. There is little space remaining for data and variables, and if more than a few (maximum seven) products are required to be analysed for example, some of the initialisation routines can be deleted to release memory for up to 18 products.

The program is complex and the author has been partly successful in the attempt to make it easy and convenient to use. Each INPUT is prompted on the screen, and error trapping, although making data entry rather laborious, goes a long way towards preventing erroneous entries. A thorough and detailed user manual is supplied to explain the program and give examples, as well as giving some background information on the retailing business. Terms are explained in the text, but as a beginner I would have liked a glossary as well for easy reference. The manual layout leaves a little to be desired as the sequence in which paragraphs are intended to be read is not always clear.

A blank data input record form is provided so that the data can be collated beforehand. On the reverse side of the cassette is a spoken introduction to the system.

Conclusion

This software should be effective in giving the retailer some quantitive assistance in assessing the effect of price changes and advertising on his business; facets of his enterprise often considered more of an art than a science at present. It will certainly make the retailer think deeply about his business. "User-friendly" aspects have obviously been given careful consideration but the program still requires considerable dedication to master. With perseverance both the novice and the experienced retailer should have something to learn from it. It shows how even the ZX81 can be used effectively for serious applications.

MICS offer a consultancy service, and a maintenance contract, to back up their software.

RETA is available from Michael Cox Information Services, 62 High Road, North Weald Bassett, Essex.

POKEing into the Spectrum display

For those of us who were used to POKEing fast-moving graphics into ZX81 display file, the organisation of the Spectrum display takes a lot of getting used to. Robert Erskine, head of the Cambridge-based software firm Microgame Simulations suggests it is not as bad as it seems.



A lot of people must have felt a sudden cold chill after unpacking their brand new Spectrums and turning to page 164 of the user manual. Not only, it seems, has Mr. Sinclair decided to chop individual characters in the display file into tiny pieces and sprinkle them liberally over the screen but the screen itself has been divided into three separate blocks of eight lines each.

Before finding out whether things are really as bad as they seem I would like to expose you to the full horror of the situation. If you run Program 1, which pokes a byte of eight pixels into each consecutive address in the display file, you will see how the screen is built up.

Starting at the top left hand corner, the top row of each character square is filled in until the end of line seven is reached, when a jump is made to the second row of the first character in line zero, and so on. The process is then repeated with the second and third groups of eight lines until the screen is full.

Could it be that cunning Mr.

Sinclair is planning to launch a new piece of hardware for £50 that will put all these pieces into some sensible order for us?

Solving the Problem

As it turns out, the solution is not too difficult. In fact, once you have familiarised yourself with the display map, you will find that in some ways it is easier to handle than the one on the ZX81. In the first place, the display file starts at a fixed address and stays there rather than floating about in RAM. Secondly, there is no danger of poking characters into hidden 'newline' flags which had the interesting effect on the earlier machine of destroying your entire program when they were poked. Thirdly, since each character is composed of eight bytes, each of which is capable of holding up to eight pixels, or plot points, you can poke any graphics you wish onto the screen without having to confine yourself to the twenty one user definable graphics available in BASIC. Try running Program 1 again but this time substitute INT(RND*255) for the number 255 and you will then get some idea of the infinite possibilities.

Suppose that you wished to POKE a 2x2 block of four black squares onto the screen near the top left hand corner. Program 2 illustrates clearly in BASIC how a machine code program might be designed to do this. The first address corresponds to the first row of the first character in line 1, column 1. The variable 'byte' is the number we wish to poke into this address and is read from the data statement in line 80. 255 corresponds to the binary number 11111111, which will appear on the screen as a black bar. The 'displacement' variable, which is also read from the data statement, is the number of screen addresses which need to be added to the start address to position the first POKE, which in the first instance is zero. The second displacement is 1, which corresponds to the position immediately to the right of the start address, followed by 31 for the top row of the third character, 1 for the top of the fourth and then 223 to jump back to the second row of the first character and so on. The pause in line 60 is included to allow you to see how the whole graphic is built up.

Displace Here

It is necessary to do things in this order because every displacement value must be small enough to occupy a single byte in the data file. Although this does not matter in BASIC, it is vital in our machine code program because each item in the data table is handled one byte at a time.

Program 3 loads the machine code program which will achieve exactly the effect as Program 2 but considerably faster. RAMTOP is set at 32229 and the program is loaded from 32300 onwards. The assembled program may then be executed by entering as a direct command: RAND USR 32301.

The data statement in line 70 holds the program itself whilst line 80 holds exactly the same data as in Program 2 except that the first number is 32 rather than zero, which is the total number of bytes in the complete graphic. Number 32 is initially loaded into address 32300 which during the execution of the program keeps a running total of the number of unpoked bytes remaining.

There is a valuable bonus gained by poking characters onto the screen by this method. Because each individual byte poked can correspond to any binary number between 00000000 and 11111111 and because each character is built up byte by byte, you can build up as many user definable graphics as you wish. Try experimenting by substituting other values for the 255s in Program 3.

The one big drawback however is that although your graphics can be moved about on the screen by overprinting it with zero bytes using a similar program and then shifting the start address, you will run into trouble if your graphic passes over or straddles the boundary between one group of eight lines and the next because the displacements will then be different. In a later article I shall show you a method of overcoming this problem.

Block Capitals

Finially, as a little light relief, here is a simple method of poking block colour graphics onto the screen.

The 'attributes' file in the Spectrum is happily arranged in a straightforward 32x24 format from address 22528 to 23295. By POKEing any one of these addresses you can print squares of any colour and attribute anywhere on the screen. For example, any number between 32 and 39 which is POKEd into an address will produce a green character square, whilst numbers between 96 and 103 will produce a bright green square.

Program 4 shows exactly which effects you will get as a result of POKEing every number between 0 and 255 into an attribute address. It should then be easy to adapt Program 3 to produce a machine code program which makes use of this to draw block graphics or fast changing backgrounds.

```
LD HL, 16417   First address
LD DE, 32333   Data start
LD A,(DE)      Byte count
LD (32300),A   Store
INC DE*        Next data
LD A,(DE)      Byte code
LD (HL),A      Print
LD A,(32300)
LD B,1
SUB B
RET Z          Finished?
LD (32300),A
INC DE         Next
LD A,(DE)      Displacement
LD B,0
LD C,A
ADD HL,BC      Next address
JP*            Next data
```

THE PROGRAMS

```
 1 REM Program 1
10 FOR x=16384 TO 22527
20 POKE x,255
30 NEXT x
```

```
 1 REM Program 2
 2 BORDER 0
10 LET addr=16417
20 READ displacement
30 READ byte
40 LET addr=addr+displacement
50 POKE addr,byte
60 PAUSE 25
70 GOTO 20
80 DATA 0,255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255
```

```
 1 REM Program 3
 2 BORDER 0
10 CLEAR 32229
20 LET a = 32300
30 READ n
40 POKE a,n
50 LET a = a+1
60 GOTO 30
70 DATA 0,33,33,64,17,77,126,26,50,44,126,19,26,
       119,58,44,126,6,1,144,200,50,44,126,19,26,
       6,0,79,9,195,55,126
80 DATA 32,255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255,223
        255,1,255,31,255,1,255
```

```
 1 REM Program 4
10 LET byte = 0
20 FOR x = 22528 TO 22528+255
30 POKE x,byte
40 LET byte = byte+1
50 NEXT x
```

MICROWARE – UNIT 5, ST. PETERS LANE, LEICESTER. Tel: 0533 29023

NOTE OUR NEW ADDRESS (Close to Clock Tower, Large Car Parks and Bus Station)

THE "FRIENDLY TO USER STORE" THE FIRST MIDLANDS REGION RETAIL SHOP SPECIALISING IN SINCLAIR COMPUTER ACCESSORIES

ZX81 NOW IN STOCK £49.95

AND "WELCOME" THE NEW DRAGON 32 AVAILABLE NOW £199.50

ALSO SUPPORTING SOFTWARE AND HARDWARE

Come and see our ever increasing range of Software · Hardware · Books · Magazines . Keyboards · Ram Packs · Graphic Roms · I/O Ports · etc.

Reliable Cassette Player – tested for loading/saving on ZX81, SPECTRUM, DRAGON, BBC ONLY £18.50

SOFTWARE & BOOKS FOR BBC and VIC 20.

IF THE PRODUCT IS GOOD AND THE PRICE IS RIGHT – THEN MICROWARE STOCK IT!

STOCKISTS OF DEAN ELECTRONICS and D.K. 'TRONICS **SUPERB** KEYBOARDS FOR ZX81 **AND** (SPECTRUM – D.K. 'TRONICS)

MICROWARE SOFTWARE for Spectrum

ALIEN COMMAND (16k) Testing Invaders style game making full use of Spectrum sound, colour and graphics ★★SPECIAL XMAS OFFER £4.75★★

BACKGAMMON (16k) Play against your computer in this version of the popular board game £5.00

BOTH GAMES AVAILABLE FROM SHOP OR MAIL ORDER

SOFTWARE WRITERS!

GOOD SPECTRUM or DRAGON SOFTWARE WANTED – EXCELLENT ROYALTIES

We apologise for delay in sending out catalogues – we are currently re-printing due to much new stock.

TRADE ENQUIRIES WELCOME OPEN 9.30 – 12 noon 12.30 – 5.30 pm SAE BRINGS CATALOGUE

Closed Monday

MICROWARE Leicester

Over half a million in use to date...



Protos 83

£59.95 inc VAT P&P £2.95

Protos 83 The ZX81 keyboard and enclosure. Features repeat function and shift lock key with LED indicator. Forty-two colour coded keys make reading easier. Steel, stove enamelled enclosure for ZX81 circuit board and keyboard. All makes of peripheral fit. Rubber feet. All push fit connections except power for repeat and shift lock.

£59.95 inc VAT P&P £2.95

Protos 1 The original and still the best looking, best made keyboard for the ZX81 is still available. Used by the Ministry of Defence, GEC, British Telecom, Government Departments and Ministries, Universities and teaching establishments throughout Europe. Protos 1 is now made only to order, with or without ZX81 PCB fitted. Ideal for business and educational applications.

Keyboard and enclosure: **£69.95 inc VAT** P&P £3.50

Write now for details and fitting data, enclosing large SAE. All orders acknowledged by return post.

And the ZX81 is still the best personal computer to learn on. But if you have made the decision to stick with your ZX81 you'll want a keyboard and enclosure built, not only to last, but also to look good.

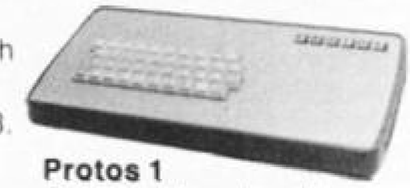
All Protos keyboards for Sinclair computers are built to the highest engineering standards, using top grade key switches and materials.

They look good too, so if you use your ZX81 in business you can take it out of the back office at last.

In the classroom Protos enclosures will stand up to the heaviest handling. Full size typewriter keys and improved key legends make computing so much easier for young programmers.

Protos 83S The Spectrum keyboard and enclosure. With left hand Symbol Shift key and audio amplifier. Keyboard switch turns amplifier on and off. Order now for delivery in January 1983.

£64.95 inc VAT P&P £2.95



Protos 1
Keyboard and enclosure

Protos Computer Systems, PROTOS

20 Ashtree Road, Frome, Somerset, UK, BA11 2SF

Formidable software

ZX Computing has a number of 'staff reviewers', but we welcome reviews from readers. This review comes from Darren-John Norbury of Andover who looks at a ZX81 cassette from ICL, and the Psion/Sinclair 'Fantasy Games' tape.

ICL's cassette G3

As it is plain to see from their advertisements in the computer magazines, the Sinclair software range is a formidable collection. All of the Sinclair tapes are made either by ICL or Psion. I intend here to look at one tape from each company.

Firstly, a look at ICL's cassette G3. Entitled 'Super Programs' the tape contains six programs, five games plus a currency conversion routine, all the programs being for 1K machines.

'Train Race' sets up three trains on the left hand side of the screen. The idea is to guess which train will get to the other side of the screen first (no prizes I'm afraid), the trains' fuel being randomly generated numbers!

The second program on the tape at least demands some skill on the part of the player. The computer asks you to choose a top value, 500 for instance, and then it selects a number between 0 and your chosen value. The player must then try and guess this number in as few attempts as possible. After each guess the computer says either too high or too low which means that, coupled with some lucky guesses, the task is not as impossible as it may sound.

Secret Message

'Secret Message' asks one user to input a message. The message is then played back when required, but upon return, is re-written in huge graphic letters which are scanned with only one or two letters on the screen at any one time. It is a very slow process so it is advised not to try the message: "Have gone round to see mother — your dinner is in the oven".

Mind That Meteor

This is, in my opinion, the best program on the tape. A meteor field is set up, in the slow manner instead of that of the slick machine code, and the player must guide his spacecraft from the left to the right of the field, which takes up only the top half of the screen, supposedly to conserve memory space.

Character Doodle

Another of those, to my mind, annoying little sketchpad programs for which I don't think I can see much utility. A bit of fun for the children I suppose.

Currency Conversion

That takes care of the five games. The last program is a currency conversion routine. I suppose it could be called a business program. I must admit that when I first ran this program I couldn't make a lot of sense out of it. This problem was mainly due to a lack of proper explanation in the accompanying cassette documentation and a similar clarity, or lack of it, in the prompts within the program. It may well drive some budding currency converters to a pencil, paper and a calculator.

Admittedly ICL's Super Program range were some of the first cassettes of ZX81 software to appear.

Conclusion

I think that this range has been left behind by some of the newer software manufacturers who are coming up with ways to put more exciting and complicated games and routines on 1K machines. Maybe ICL are working on it as I write or maybe it is felt throughout the software industry that the 1K program market is diminishing with so many users moving up to 16K after a short period with just the 1K internal RAM.

Psion's Fantasy Games

If cover illustrations sold computer cassettes then Psion's 'Fantasy Games' tape would probably be the country's best seller, in fact it may be for all I know.

The cassette has two games, one on each side, the major game being 'Sorcerer's Island' on side B. Side A contains a shorter game called 'Perilous Swamp' which is intended as a warm-up to the other longer game.

In 'Perilous Swamp' a board is set up representing the swamp area. The board is laid out in squares like a chess board. Represented in the swamp land are the player and the Princess who has to be rescued by the player from the clutches of an evil wizard and removed from the island. The player moves one square at a time, vertically, horizontally or diagonally, but cannot go through the black squares which are marsh, only on the fairly dry white squares.

At the beginning of the game the player is alloted a certain amount of life points. Every move towards the Princess involves a meeting with some sort of ghastly being who is guarding some valuable treasure. Each creature has a strength rating compatable with the player's life and so you must try to defeat the creature by numbers and so gain treasure points by depriving the monstrosities of their valuables. Of course, once all the life points have been used up . . . finito! No treasure, no Princess, no life — end of game!

Incidentally, you may think I am having trouble trying to say what I mean here. The games are so involved that it is very difficult to describe them. In the cassette flap literature it says: "Detailed instructions are not included since the player must discover the rules by playing the game".

In the first game 'fight' is only one option for the adventurer when meeting a nasty. He may also bribe the creator or simply run.

But now a warning. You are liable to get very self-confident after getting the Princess away from the island along with an amassed amount of well-fought-for wealth two or three times. Go on then, clever clogs; turn over and try the other game.

Sorcerers Island

This works on the same principle as 'Perilous Swamp'. It's just that the second game is on a much grander scale.

Okay, I'll let out a little secret. Much as I have tried I find escaping from the island an impossible task. This game is a challenge though. It's not the sort of game you tend to give up because it's boring or it can't be done. It can be done, I *know* it can — I think.

My personal opinion of 'Fantasy Games' is that it's in a class of its own from a point of addiction and absorption.

The games are not over quickly and, although there is a great element of luck involved, it is quite possible to spend an evening thinking one's way off of 'Sorcerer's Island'.

ICL Super Programs Cassette G3 is priced at £4.95 while Psion's Fantasy Games costs £4.75.

Now, it IS possible to get off of this island . . . I'm sure it is . . . isn't it?



Alistair Lindsay from Frodsham, Warrington reviews Bridge Software's "Galaxy Invaders"

After hearing so many good reports about this cassette, I decided that I must see it. I sent off on a Bank Holiday and it arrived four days later — good service, you must admit.

When you have loaded it, (I did it first time), it runs automatically. You are asked what level you want to play on, between 0 and 9. On level 0 it is just about possible to have a long game, but you have to be a complete expert to score anything on level 9. The game is flicker-free but that is replaced by the fact that the aliens "shimmer". This means that it is hard on the eyes and therefore difficult to play for longer than half an hour. Apart from this fact, the game is very good and I can see that much more thought has gone into it than into some (most) ZX Invaders. The shapes of the invaders are very good, considering that they can only use the Sinclair character set.

One thing which I discovered is that by using Tim Hartnell's method of finding out how much memory a program uses, this Invaders takes up just over 2K. In some of the advertisements it says that a 16K RAM pack is required. Bridge Software are not the only company to do this, by any means. (Tim Hartnell's method is: "Print Peek 16396 + 256 * Peek 16397 — 16509" When this is used as a direct command, it will print the number of bytes which that particular program has used).

Conclusion

This is definitely one of the best "Invaders" type games on the market at the moment for the ZX81. It only takes up just over 2K of memory, so do not be put off if you do not have a full 16K expansion. "Galaxy Invaders" costs £3 from Bridge Software of 36 Fernwood, Marple Bridge, Stockport, Cheshire SK6 5BE. It has adequate instructions and my percentage rating for it (with the average "invaders" scoring 50%) is 70%.

SQUEEZING IT ALL INTO 1K

Many articles have been published about saving precious bytes on the 1K ZX81. Adam Waring has delved deeper into the subject than most, and here presents a summary of all the worthwhile memory-saving hints. He illustrates his discoveries with three programs which show just how much can be squeezed into the unforgiving RAM.

Every time a number is used in a BASIC program 7 bytes of memory are used up. You can think of a BASIC program as: 7 times the amount of times numbers appear, plus the number of characters per line, plus 5. Note that the line number is not counted in either the amount of numbers or the amount of characters. The 5 at the end of the line contains the line number. It also holds the length of the line and the end of line NEWLINE character.

It is obvious that the fewer numbers in a program, the more memory you save. It is feasible to type:

```
10   LET X = VAL "5"
20   LET Y = CODE "*"
30   LET Z = INT PI
```

Instead of:

```
10   LET X = 5
20   LET Y = 23
30   LET Z = 3
```

Line 10 saves you 3 bytes, line 20 saves you 4 bytes, and line 30 saves you 6 bytes. Each line saves you only a few bytes, but added together save you 13 bytes. This is the method that most articles I have seen show. It allows you to add a couple of lines to a program that would, under normal circumstances run out of memory, but if your program is pretty complex or big (for 1K) then you need a way to save memory a lot more efficiently than 4 bytes a time. The ZX81 saves all variables, as well as the program, on tape. If you initialise a variable in command mode (without a line number) then no memory is used as the line doesn't appear in the program (though memory is used to store the variable, which would be used in the normal way, anyway). Typing:

```
LET X = 5
LET Y = 23
LET Z = 3
```

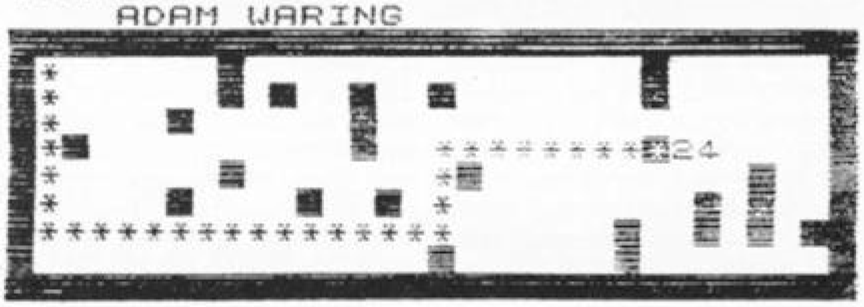
Remember, no line numbers. Saves you a total of 49 bytes! Unfortunately, there is a snag, as always, and that is you can't run programs! It is possible to use GOTO 1 instead. That's not too hard to remember, and if you use this method, you soon get used to it, but if someone loads a program from tape then they will instinctively type run. It is best, therefore, to include a LOAD and GO routine in a program. By this I mean type:

1 SAVE "PROGRAM NAME"

With the program following. It is best to use just 1 letter for the program name. When LOADed the program will automatically continue from the next line.

If we take the memory saving technique of mine a step further, we can replace every number, even GOTO's and similar statements, with a variable. I have included three example programs for you to key in. They should give you an idea of what you can really do with your £49.95 lump of plastic. The program is listed, which you should type in first, with the variables following...Don't forget, use GOTO, not RUN. Enough of the chat, let's see what can be done.

TRAPPED
ADAM WARING



Trapped

The game of Trapped follows the maze type theme with a difference; in this version it is impossible to win! You can only do the best you can. The idea is to navigate your "*" around the maze and get as far to the right of the screen as possible. '6', '7' and '8' send you down, up and right respectively. Left is ommited as in practice it was never used.

PROGRAM LISTING

```
   1 SAVE "■"
  20 INPUT A$
  30 CLS
  40 LET A$=""
  50 LET A=B
 100 FOR N=A TO C
 110 PRINT AT A,N;"■";AT D,N;"■"
;AT N/E,C;"■■";AT RND*I,RND*C;"■
"
 150 NEXT N
 180 LET A=NOT A
 190 LET N=A
 245 LET B$=A$
 250 LET A$=INKEY$
 260 IF A$<"6" OR A$>"8" THEN LE
T A$=B$
 270 LET A=A+(A$="6")-(A$="7")
 280 LET N=N+(A$="8")
 300 PRINT AT A,N;
 310 IF PEEK (PEEK 16398+PEEK 16
399*256)=G THEN GOTO H
 320 PRINT "*"
 350 GOTO F
 500 PRINT "■";N
 530 GOTO D


VARIABLES
 LET B=0
 LET C=31
 LET D=9
 LET E=4
 LET F=210
 LET G=128
 LET H=500
 LET I=8
```

Shoot-Out

Most people enjoy a game which involves shooting things. Shoot-out gives you this pleasure. Be careful though, a stray bullet from your gun could mean curtains — for you! 3 types of thingees, all from the outer reaches of the galaxy, hover above you. The thingees with the uncanny resemblance to the letter "Q" are the baddies. Indeed, pumping lead into one of these gains you a point. Don't hit the "X" shaped beings though, as these are your friends. Hitting these soon runs down your score, and quite rightly too. The enemy are not without defence though, space mines, or more accurately, "*"s, are up there too. A carelessly aimed shot, hitting

one of these spells "KABOOM" and shortens your life drastically. A game usually lasts one minute exactly. The real time clock inside the deepest recesses of the ZX81 enables us to be so accurate. Keys '5' and '8' move your gun left and right. '0' fires it. This program shows just what can be fitted into the 1K ZX81 if the program is thought out carefully enough. Enough memory is saved to have 5 different ratings at the end of the program, from 'Pathetic' to 'Champ'!!

```
SHOOT-OUT
     ADAM WARING

  Q    X          X     *        Q     X



                             ▟
     SCORE=14
EXCELLENT
```

PROGRAM LISTING

```
   1 SAVE "A"
  20 LET S=F
  50 POKE B,D
  60 POKE C,E
 100 PRINT AT F,RND*G;"   Q   ";AT
 L,A;"  ▟  "
 120 IF RND>I THEN PRINT AT F,RN
D*G;"  X  ";AT F,RND*G;" * "
 150 LET A=A+(A<G AND INKEY$="8"
)-(A>F AND INKEY$="5")
 160 IF INKEY$="0" THEN GOSUB D
 200 IF PEEK B>J THEN GOTO K
 205 PRINT "EXCELLENT" AND S>H A
ND S<U;"CHAMP" AND S>=U;"OK" AND
 S>=L AND S<=H;"RUBBISH" AND S<L
 AND S>=Q;"PATHETIC" AND S<Q
 230 INPUT A$
 240 CLS
 250 GOTO H
 255 FOR M=R TO Z STEP -Q
 310 PRINT AT M,A+Z;"█";AT M,A+Z
;" "
 330 NEXT M
 340 IF PEEK (P+A)=F THEN RETURN

 345 IF PEEK (P+A)=N THEN GOTO O
 347 IF PEEK (P+A)=T THEN LET S=
S-Q
 350 PRINT AT M,A+Z;"▩";AT M,A+Z
;" "
 370 LET S=S+Z
 375 PRINT AT H,Q;"SCORE=";S;"  "
 380 RETURN

VARIABLES
 LET A=9
 LET B=16437
 LET C=16436
 LET D=255
 LET E=183
 LET F=0
 LET G=29
 LET H=10
 LET I=0.7
 LET J=243
 LET K=100
 LET L=9
 LET N=23
 LET O=205
 LET Q=2
 LET R=8
 LET T=61
 LET U=15
 LET Z=1
 LET P=PEEK 16396+PEEK 16397*256
+2
```

Invaders

Every fun loving ZX81 owner dreams of having a really good space invaders program to show off to their friends. Unfortunately these cost about £5 a throw so you'll have to settle for this instead. Most basic invaders programs I have seen have only one invader who doesn't know the meaning of hostile! So this version has something going for it. There are 8 invaders, all thinking cruel thoughts as they attack the almost defenceless earth. As laser bombs fall unmercifully from the sky, your laser canon does its best to keep out the way. The green (or in this case, black) meanies do not move from side to side as is the tradition — it would be far too easy to win. Instead they move randomly. If you eliminate all the invaders then a new sheet instantly pops up. The keys used are the same as in the previous program — 5, 8, and 0. In this version the aliens do not advance, you have a time limit instead. Well — even with this method you can't achieve the impossible — or can you?

```
INVADERS
     ADAM WARING
                █ █          █    █




                                  $
 SCORE=4
```

PROGRAM LISTING

```
   1 SAVE "B"
  10 LET A=T
  20 LET S=G
  30 LET B=A
  40 LET A$=" █ █ █ █ █ █ █ █ "
  90 FOR N=H TO I
 100 LET A=A+(A<K AND INKEY$="8"
)-(A>G AND INKEY$="5")
 140 PRINT AT G,B;A$;AT T,A;" $
"

 160 IF INKEY$=STR$ G THEN GOSUB
 L
 170 GOSUB W
 173 LET B=B+INT (RND*J)-H
 175 LET B=B+(B<G)-(B>T)
 180 IF A$="                    " T
HEN GOTO O
 200 NEXT N
 230 INPUT A$
 240 CLS
 250 GOTO R
 270 LET U=INT (RND*T+B)
 290 IF PEEK (P+U-H)=G THEN RETU
RN
 300 FOR M=H TO T STEP J
 310 PRINT AT M,U;"█";AT M,U;" "
 330 NEXT M
 340 IF U-H=A THEN GOTO V
 345 RETURN
 350 IF PEEK (P+A)=G THEN RETURN

 360 LET A$(A-B+R)=" "
 370 LET S=S+H
 375 PRINT "SCORE=";S
 380 RETURN

 LET G=0
 LET H=1
 LET I=60
 LET J=3
 LET K=29
 LET O=40
 LET R=2
 LET T=13
 LET U=14
 LET V=210
 LET W=270
 LET L=350
 LET P=PEEK 16396+PEEK 16397*256
+2
```

Amateur radio will become much clearer after 3rd Dec.

The radio market has become more complex. Things have become more confused.

Wires get crossed as new equipment floods onto the market.

At the end of the day, even the most avid enthusiast spends more time trying to find out about new equipment than on the airwaves using it.

As for the novice?

They stand little chance of picking anything up at all.

So we've decided to clear things up.

On December 3rd our new magazine Ham Radio Today begins.

Not a magazine you need a degree in electronics to decipher.

Or one that still calls your gear a wireless.

Rather a magazine that simply clarifies the vast range of electronic gadgetry available.

Lists new equipment, analyses its performance.

Thorough reviews, special features, news items and constructional projects.

In a clear and concise way that will give everyone a perfect 5 + 9.

Ham Radio Today.

Tomorrow . . . tune in and find out, 73.

HAM RADIO TODAY

Published by
Argus Specialist Publications Ltd.

Getting into print properly

Tim Langdell looks at Capital Computer's print interface which allows you to add 'proper' printers to the ZX81, Data-assettes tape control system ZX99, and the Z-XTRA by Cirrus which provides a 16K RAM extension, and a series of extra facilities in an EPROM.

Whilst the Sinclair printer is clearly excellent value for money, the quality of its print could hardly be described as high quality. So if you are interested in producing listings that are clear to read or in doing some kind of word-procssing with your ZX81 then a more standard type of printer would be useful to you. Up until recently, though, it was not possible to add such a printer onto a ZX81. The printer interface from Capital offers this facility for attaching either a serial (RS232) printer or a parallel (centronics type) one. The interface comes as two printed circuit boards. One is a mini-motherboard with two slots for hardware add-ons. The printer interface card itself fits into one of these slots and the other is free for another add-on of your choice. The motherboard also duplicates the connector at the rear of a ZX81 and so you can still attach to 16K RAM pack quite easily.

The edge connectors on the boards were of reasonable quality and gave a snug fit. However, I would not advise moving the ZX81 around much on a table top when the interface is attached. The interface allows a centronics or RS232 type printer to be attached because it has an EPROM on board (a 2K 2716 one by Mostek) which contains the necessary software to decode the output from the ZX81 into the signals which these two types of printer need to receive. Thus this software in EPROM handles the transmission of the characters to be printed by dealing with the busy signals, strobes, and setting the necessary Baud rate (speed at which the characters in the form of data bytes are sent to the printer). This EPROM sits in memory above the ZX81's ROM at around 9K (from 2000 to 2800 HEX). Therefore this device is unlikely to conflict with any other add-ons and does not interfere with memory extensions up to 48K.

Setting Up The Interface

This was reasonably easy, although the instructions which came with the boards could have been more explicit. For a serial printer you would need a 14 pin DIL plug and a 25 D-connector joined by ribbon cable of at least 6-way. On the board are two DIL sockets, one 14-way and the other 16-way. The 14-way is used for a serial printer and the connections area as follows:

8 N/C	7 GND
9 N/C	6 N/C
10 N/C	5 N/C
11 N/C	4 N/C
12 I/P TTL	3 I/P RS232 (BUSY)
13 O/P TTL	2 O/P RS232
14 GND	1 GND

As you can see outputs for TTL devices are also catered for. Capital claim this to be a full specification RS232, which will therefore amply meet most people's needs to add printers or modems.

The baud rate for the RS232 is set by a DIL switch on the PCB. Switches 5 to 7 can be set either on or off to achieve baud rates from 50 to 9600. The booklet with the interface makes it clear how to do this.

The Printer

The parallel printer is as easy to attach and requires a 16 pin DIL header, a standard Centronics plug and 16-way ribbon cable. The connections to the 16 pin DIL socket on the board are:

9 D7	8 GND
10 D6	7 FAULT
11 D5	6 N/C
12 D4	5 UNIT SELECT
13 D3	4 PAPER BUSY
14 D2	3 BUSY
15 D1	2 ACKNOWLEDGE
16 DO	1 DATA-STROBE

Selecting the number of characters per line on the print out is done by POKEing location 16507 with the required number. If you don't tell the interface otherwise it POKEs this location with 64. You'll note in your ZX81 handbook (p.179) that this location is free in the systems area of RAM. Calling the main printing routine is easily done by having a programme line or direct command as PRINT USR 9407. An asterix appears prompting you to type in either 'S' to select the RS232 output, 'P' for the parallel printer, 'K' to print directly from the keyboard (note this isn't possible with Sinclair's printer), or STOP to return to the BASIC programme again and halt printing. You can also get a hexadecimal memory dump by pressing 'M' in response to the asterix prompt. But first you must set up which area of RAM/ROM you want to print to the printer by POKEing the following locations:

START OF DUMP –
  POKE 16434, low byte
  POKE 16435, high byte
END DUMP –
  POKE 16438, low byte
  POKE 16439, high byte

Another look at the systems area on pages 177 to 179 of your manual will reveal that these locations are usually used by the ZX81 as the SEED for RND and as the coordinates for the last PLOTed point respectively. Clearly these aren't used during a print out for these purposes.

Variables can be printing during a BASIC program by using a few lines like:

```
10 CLEAR
20 LET A$ = "STRING"
30 LET A = USR 9533
```

LLIST will list a BASIC program beginning at the line where the cursor is, much as with the Sinclair printer. LPRINT works as usual too and sends the first string in the variable area. SHIFTED 9 allows (to get inverse 'G' cursor) lower case letters graphics or control commands. For instance, SHIFTED A gives the Centronics command DC2, SHIFTED S gives the 'SO' command and so forth. If you do not have a larger type printer then these commands will not mean much to you. If you have a printer though, I suggest that you consult its manual for a full explanation of the control commands.

Interfaces

Capital claim that the interface has been updated since our sample of it, and now the EPROM offers more potential. Details are not available at present, but we understand the price is about the same at £39.95 + VAT. The mini motherboard costs £14.50 + VAT. If you require a Centronics or RS232 type interface, then this one from Capital certainly does the job. At a total of about £45 it is not cheap, but nonetheless offers good value for money. Capital seem committed to keeping their customers informed of improvements, and mention such tempting items as a battery back-up memory module and an eprom programmer to be on the way. Out of interest, the mini-motherboard also has a reset button on it which can be very

useful if your programme crashes. Using a reset button is much kinder to the machine than switching off and on again.

ZX99 from Data-assette

The ZX99 is a tape control system allowing you to handle up to four cassette recorders with your ZX81. It also has a simple version of an RS232 interface to allow a full size printer to be attached. The ZX99 comes in a black casing which is well made and well designed to match the ZX81's styling. It connects to the ZX81 with an edge connector of good make, and the '81's rear connector is duplicated at the rear of the ZX99.

The sockets on the sides of the ZX99 allow the four cassette recorder to be attached and the RS232 as well. There are four LEDs on the front panel which indicate which of the cassettes is in operation. The unit comes with one of the most complete and lengthy manuals I have seen with any ZX81 add-on. It has some 48 typed pages of information. The four cassettes which can be attached are intended to be two for input and two for output. That is, two are intended to be used as additional data storage and two to write programs or data on. The ZX99 contains a 2K EPROM which does all the work. Like Capital's interface this device also sits in memory at the 2K above the Sinclair 8K ROM.

The unit is not a simple data storage device, but has many functions which are called with USR commands. There are four major types of command available: selection of a tape drive, reading, writing or skipping blocks of data; copying tapes; and printing data or listings via the RS232 interface. It is possible to select two outputs at once and hence make two copies onto two different cassettes of the same data or program. The use of storing data on tape means that by saving it in blocks you can load data block by block until the data you require is in memory. Thus, whereas a 16K memory extension would probably only allow you to handle a list of about 150 names and addresses, use of the ZX99 would let you store and handle many times this number. The ZX99 doesn't use the ZX81's LOAD to get blocks of data into RAM as this would destroy the program in the memory as loading took place. The ZX99 uses a buffer by requiring you to send data to it in the form of strings. The string can have any label in the ZX81's repertoire, but it is signposted by the contents of Z$. So if you choose to call your buffer A$ and have it 200 bytes long then a program would include these lines:

```
  10  DIMA$ (200)
1000  LET Z$ = "A"
1010  LETSTATUS =
USR 8210 (reads data to tape from A$)
```

Data-assette recommend the use of the form 'LET STATUS = USR. . .' rather than 'RAND USR. . .' because the value returned to the variable 'STATUS' has been designed to give feedback as to the types of errors that might have happened. Reading into the string A$ (or whatever) is just as easy using USR 8213. Many examples and suggestions about handling data are given in the ZX99's manual. The unit also has a block skip instruction which allows you to get the ZX99 to skip past each program to check what you have on any particular tape, and to get accurately to a space on tape for the next program to be saved to. The ZX99 can also be used to copy tapes — either making one new copy or two depending on which USR command is chosen. The manual's section on using the RS232 interface is extensive and shows clearly how baud rates from 110 to 9600 can be selected. The interface also allows choice of so called 'stop bits' and parity checks. Inverse video produces lower case letters on the printer, but the ZX99 gives you the option of reversing this so that the lower case letters are normally produced and the graphics mode of the '81 is then used to get capitals. This would be very useful when doing word-processing with an expanded ZX81. Error report codes are listed at the back of the manual (the ZX99 produces some extra codes like the ZX81's own ones which have such meanings as insufficient memory space, or BREAK pressed while ZX99 in progress). There is also a full listing of the completion codes which are returned when a USR function is called with 'LET STATUS = . . .'. These codes allow a very detailed analysis of why a recording may have failed — anything from detecting poor electrical contacts to failing to select the correct channel. All in all, it is a very impressive device for those who feel that they would like to have some of the data storage and handling facilities ususally associated with a disc drive, but cannot afford such a system, (very few exist for the ZX81 anyway). The problem is, how many people require this type of storage? Now that 48K and 56K RAM packs are available one would imagine these would handle most requirements of ZX81 users. The tape system is very slow on the ZX81, as those of you fortunate enough to compare it with the faster Spectrum system will now be painfully aware. Thus, while it would allow you to handle very large address lists and so on, it would be rather frustratingly slow. Indeed it is hard to imagine such purposes being seriously achieved by a ZX81 rather than a faster machine such as an Atom, Apple, BBC, or even the Spectrum. To be fair to Data-assette though, when they launched this device the ZX81 was the only cheap micro available.

Z-XTRA by CIRRUS

This device is a late entry to the scene of ZX81 add-ons. It is both a 16K RAM extension and a series of extra facilities resident on an EPROM. The unit is a large black metal box about twice the size of the ZX81 itself. Connection is made to the ZX81 with a standard edge connector — although this is probably the best quality one I've seen on any ZX81 add-on. The Z-XTRA plugs into the mains and provides power for itself *and* the ZX81. Therefore the ZX81's own power supply is not needed. On one side of the Z-XTRA is a 2.5mm socket to take a plug from your cassette recorder's 'remote' socket. This allows the Z-XTRA to turn the cassette's motor on and off when necessary. There is also a reset button, which allows you to start the ZX81 again after a program crash without resorting to



The ZX99 in use, controlling four cassette machines:



The ZX99:

the extreme of switching the power supply off and on.

On switching on you will have 16K of RAM and to access the Z-XTRA's facilities you will need to type RAN USR 9834. (the EPROM resides once more at the 2K above the 8K ROM). A menu may then appear on the screen. I say *may* because at the time of writing CIRRUS had not yet fully decided which options would be available with the basic machine. However, with all options chosen a display like this will be obtained:

1. LOAD
2. SAVE
3. HEX DISPLAY & MODIFY
4. INVADERS
5. FREE MEMORY
6. DISABLE KEYBOARD BEEP
7. ENABLE KEYBOARD BEEP
8. SWITCH TAPE ON
9. SWITCH TAPE OFF

CIRRUS were at pains to express that this is not the choice available on the basic unit, but rather a 'taster' of what will be available to purchasers of the unit in the form of custom EPROMS fitted in the Z-XTRA's case. As it is when you buy it it will probably only have the so called 'Z-XPRES' tape SAVEing and LOADing system in it (no.s 1 and 2 above), so I'll mention this first.

SAVEing and LOADing

Z-XPRES is a very fast program/data loading and saving system which allows you to save or load the entire 16K or RAM in less than a minute (typically 30 seconds). This gives it a baud rate of several thousand and makes it even faster than the Spectrum. However, you have to save the entire 16K even if there's only a 1 or 2 thousand byte program in it. This is not a problem though as the saving and loading is extremely fast. There is also a 'checksum' on the operations meaning that loading and saving errors are detected and poor copies are very unlikely. If a LOAD is terminated by 0/0 then the loading went well, but if it returns X/0 then there was an error in the loading. Note that unlike the ZX81 or Spectrum's normal loading systems this system will load a poor program and allow you to list it and make corrections.

In general the facilities available with my review sample were impressive. It was possible to obtain a HEX display of memory contents by simply entering '3' when the menu was displayed. SLOW mode must be chosen, and an address prompt appears. The address in HEX where you wish to start is entered and when SPACE is pressed the memory contents are displayed filling the screen. A cursor indicates which byte is presently being looked at by the unit, and this cursor can be moved through memory very quickly, (pressing the SPACE moves it right, N moves it left). The screen automatically scrolls up or down depending on where the cursor is. To copy the screen to a ZX printer you simply press 'Z' and to exit the facility you press 'Q' (for 'quit'). To modify any byte in RAM (not ROM of course) you simply align the cursor with the byte in question and enter the new value. Pressing the SPACE again moves the cursor on and leaves the new byte in RAM displayed on the screen. This facility has obvious uses to the machine code programmer, but is perhaps of rather less use to the rest of us.

Memory

Free memory could be determined by selecting '5', and this displayed the free memory up to 9999 bytes. Above this value the most significant byte was displayed in HEX and the least significant in decimal — a rather puzzling method. Thus having 15240 bytes left produced F240. To get a fully decimal version you could enter PRINT USR 8465.

When the unit was switched on pressing any key produced a beep.

To turn off the beep '6' could be selected when in the menu mode, and '7' would switch back on again. The beep sound could be switched on at any time by RAND USR 9716, and off again with RAND USR 9735. The beep sound could also be sounded for a set period using RAND USR 9779. The period of the beep is set by entering RAND x beforehand, where x has a value between 0 and 255. This beep period uses just the low byte of the RAND seed and so a maximum beep of 25.5 seconds is available.

It was possible to turn the tape recorder on and off by selecting either 8 or 9 in the menu. A memory test was also possible by entering PRINT USR 9023,PEEK(16417). This tests every bit pattern for every location and takes about 1 min 40 secs. I'm not sure whether many ZX81 users would find this very useful. A more useful facility was the RENUMBER routine called with RAND USR 8192. The version in EPROM was only a prototype renumbering but leaving GOTOs and GOSUBs unchanged. A full version will apparently become available.

To Sum Up

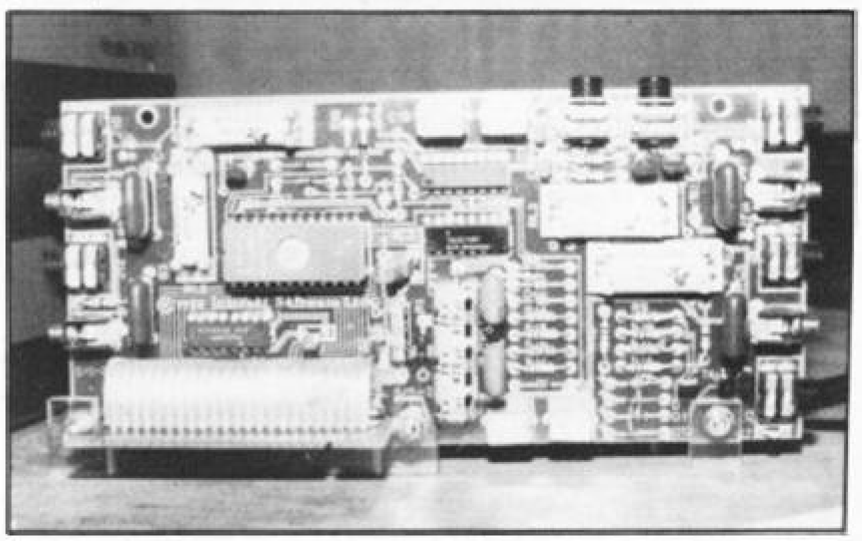
The Z-XTRA consisting of Z-XPAND (a 16K RAM extension), Z-XTEND (an EPROM) incorporating Z-XPRES (only) — a high speed cassette I/O port, costs £59.95 in kit form, and £79.95 fully built. Postage and packing is an extra £2.95. At this price I have my doubts about how many ZX81 owners will buy this unit. You may already have a 16K RAM pack in which case you might be more interested in the EPROM facilities only. Cirrus do not plan to offer this though, which is a pity. Perhaps they will change their minds. If the 16K RAM is not needed then all you are left with for about £80 is a reset button, a fast cassette SAVE and LOAD and a I/O port. Although Cirrus neglect to mention it in their literature, there is a connector at the rear of the unit taking the output of a Z80 PIO chip to enable an EPROM programmer to be added. At £80 this does not seem good value. Their choice of only selling the Z-XPRES with the unit seems a pity, why not a few extra useful routines such as a RENUMBER or the BEEP facility too? They had a space invaders game on the EPROM for demonstration purposes, and so clearly intend to market such software on ROM too. Perhaps a game on the EPROM as it comes might be worthwhile. The total package with all of the options reviewed above is estimated by CIRRUS to sell for £70 in kit form and £93 ready built. This is a great deal of money to spend on a £50 machine!

In conclusion this is a nice piece of equipment which is well made, but is likely to be outside the price range and needs of the average ZX81 owner.

ADDRESSES:
Capital Computers Ltd.,
100 Church St.,
Luton, Beds LU1 3LG

Data-assette,
44 Shroton St.,
London NW1

Cirrus Electronics,
Mazzard Cottage,
East Hill,
Ottery St Mary,
Devon EX11 1QH


Inside the ZX99:

CPL COMPUTING

ZX SPECTRUM

SOFTWARE

DEFENDER Version of the popular arcade game, including; six types of alien; full vertical movement and two dimensional play; high score; smart bombs.

MISSILE DEFENCE Fire your intercept missiles, protecting your cities and missile bases from the enemy. With on screen scoring.

SPECTRUM INTRUDERS Includes an ever increasing rate of difficulty. High scoring saucers and continuous fire.

SPACE WARS Two player game; destroy your enemy before he destroys you. Complete two dimensional movement.

All programs include two dimensional animated graphics, with full colour and sound.

ANY 2 PROGRAMS...................... **£4.95**
ALL 4 PROGRAMS....................... **£7.45**

Send to: **CPL COMPUTING, Software Department White Lodge, Skidby, N. Humberside HU16 5TF.**

Make all cheques/postal orders payable to: MR C. DORAN

MAIL ORDER ONLY

THE BUFFER

MICRO SHOP

(NEXT TO STREATHAM STATION)

★★★

THE OLDEST SOFTWARE SHOP EXCLUSIVELY FOR

SINCLAIR COMPUTERS

PROGRAMS, GAMES, "ADD/ONS"

★★★

SPECTRUM SOFTWARE NOW AVAILABLE

★★★

MOST OF THE MAIL ORDER ITEMS ADVERTISED IN THIS MAGAZINE AVAILABLE OVER THE COUNTER

★★★

LOADING PROBLEMS? TRY OUR INTERFACE
BUSINESS & TECHNICAL DATA HANDLING PROGS.
PROPER KEYBOARDS: CONSOLES: VDU's

★★★

374A STREATHAM HIGH ROAD, LONDON SW16
Tel: 01-769 2887

S.A.E. APPRECIATED FOR CATALOGUE

OPEN TUES-SATS 10.30 - 5.30. CLOSED MONDAYS.

ZX81 GAMES

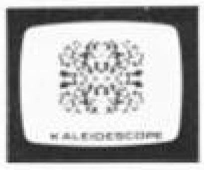
from

J.K. GREYE SOFTWARE LTD

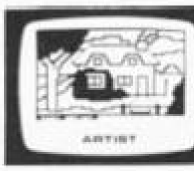
THE NEW GENERATION SOFTWARE HOUSE

"Without question the finest machine code games available today"......... J.N. ROWLAND Product Manager for W.H. SMITH.

GAMESTAPE 1 for 1K ________ only £3.95
10 Games incl. ASTEROIDS, UFO, CODE, BOMBER, GUILLOTINE, KALEIDESCOPE, etc.
PROBABLY THE BEST VALUE 1k TAPE AVAILABLE.



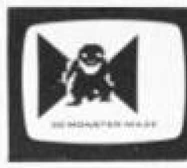
We've done in 1k, games which some of our competitors require 16k to do!



GAMESTAPE 2 for 16K ________ only £3.95
***STARFIGHTER** Superb machine code Space Battle. Set against a background of twinkling stars, with stunning explosions — if you can hit the enemy!
PYRAMID Can you move the Pyramid? Make a mistake and it will collapse! A Thinkers game.
ARTIST The ultimate Graphic Designers aid. 8 Directions, 10 Memories, SAVE, COPY, RUBOUT, CLS, etc.

GAMESTAPE 3 for 16K ________ only £4.95
***CATACOMBS** A Multi-Level Graphics Adventure. Each level can contain up to 9 Rooms, 8 Passages, 7 Monsters, Food, Gold, Traps, Phantoms, an Exit (to the next level), and there's an infinite number of levels.
NOTE. . . . This is NOT one of the necessarily limited text Adventures as sold elsewhere.
"An excellent addictive game which will keep you amused for hours." . . . . COMPUTER & VIDEO GAMES.





GAMESTAPE 4 for 16K ________ only £4.95
***3D MONSTER MAZE** The Game to Top All Others. Unbelievable Graphics! Can you find your way through the Maze? The EXIT is there somewhere, but then so is a T.REX, and its after YOU! All in 3D (the T.REX will actually run towards you in full perspective!), you've never seen anything like this before!
"3D MONSTER MAZE is the best game I have seen for the ZX81" . . . COMPUTER & VIDEO GAMES
"If I had to choose just one programme to impress an audience with the capabilities of the ZX81, then J.K. Greye's 3D MONSTER MAZE would be the one without doubt" . . . . ZX COMPUTING. *"Brilliant, brilliant, brilliant!"....POPULAR COMPUTING WEEKLY*

GAMESTAPE 5 for 16K ________ only £4.95
***3D DEFENDER** The Ultimate Space Game. Super fast Machine Code 3D version of the Arcade favourite. You have to save your home planet from the marauding Alien Spacecraft. This is all in 3D, your viewscreen shows you the view out of your fighters cockpit window. The backdrop moves when you turn, or fly up or down (8 flight directions), just as if you were really flying it! But then YOU ARE! The Enemy Saucers will actually zoom towards you in 3D, and shoot you if you let them! Your display includes Score, Shield Strength, Altitude, Proximity, Forward Radar and your viewscreen, which shows your rotating home planet, backdrop of Stars, Meteors, Explosions, Plasma Blasts, your Photon Beams, up to 4 Enemy Saucers and of course its all in full 3D!
"Another 3D winner".............................. *SINCLAIR USER*





GAMESTAPE 6 for 1K ________ only £1.95
***BREAKOUT** Super Fast Full Screen Display Game. Your all time favourite with an added twist. See how much Money you can win and watch the pounds convert to Dollars. All in Machine Code for Fast Action with 3 Speeds, 2 Bat Sizes and three angles of rebound! The best BREAKOUT around and at this price you can't go wrong!
"The best of its kind" .... *WHICH MICRO & SOFTWARE REVIEW*

GAMES MARKED * INCL. MACHINE CODE.
Prices include VAT and U.K. P. & P.
(Add appropriate Postage on Foreign Orders). Cheques/P.O.s to

J.K. GREYE SOFTWARE LTD

Dept ZX16, Brendan Close, Oldland Common, Bristol BS15 6QE

CREDIT CARD SALES: FOR INSTANT DESPATCH, **BY PHONE ONLY**
TEL: 01-930-9232 (9 am - 7 pm)

If you prefer to see before buying, our range of **GAMESTAPES** are stocked by the following stores.

BUFFER MICROSHOP	374A Streatham High Rd., London SW16:
GAMER	24 Gloucester Rd., Brighton;
GEORGES	89 Park St., Bristol, Avon;
MICROSTYLE	29 Belvedere, Lansdown Rd., Bath, Avon;
MICROWARE	131 Melton Rd., Leicester;
SCREEN SCENE	144 St. Georges Rd., Cheltenham, Glos;
W.H. SMITH	Over 200 Computer Branches;
ZEDXTRA	5 School Lane, Kinson, Bournemouth, Dorset:

TRADE & EXPORT ENQUIRIES WELCOME

HAVEN HARDWARE ZX PRODUCTS

SAE for details: 4 Asby Road, Asby, Workington, Cumbria CA14 4RR.
Prices include VAT and P & P.
Some of our products are also available from the following agents:
THE BUFFER SHOP, 374A Streatham High Road, LONDON SW16
MICROWARE, 5 St Peter's Lane, LEICESTER
PHILIP COPLEY HI-FI, 7 Clifford Court, Ossett, WEST YORKSHIRE

SWAMP MONSTERS (NEW) £6.50 A fantastic high speed game in machine code with full colour and sound. Guide your robot through an alien swamp and try to destroy the monsters that inhabit it. (Model B or 32K Model A + User Port)

SHOOTOUT £5.00 Pit your wits against the western gunfighter out to kill. Run and dodge behind the wagons and cactus whilst shooting at this killer. Full colour graphics and sound.

INVADERS £6.50 A fast moving space game, compiled in machine code. It utilises Mode 2 colour graphics and sound.

FIRIENWOOD (NEW) £6.50 Journey on a quest for the Golden Bird of Paradise through caverns and a forest in a land of Monsters and Magic where death waits around every corner.

SPACE FIGHTER £8.50 The Galaxy is invaded by alien fleets as you fly your ship across the empty wastes of space in this excellent game. Will you survive this very complex test of your ability.

HANGMAN £4.00 A colourful and entertaining version of this well known word game. Three levels of play against the clock. As you improve your score the response time is reduced.

PROGRAMS Deduct 10% off list price for three or more program tapes ordered. Send S.A.E. for full range of programs and price lists. A 32K memory required unless marked *.

WANTED URGENTLY

Bulk supplier of 5-way keyboard sockets as used on ZX81 and ZX SPECTRUM required. Please send samples of 100 off, 1000 off and 10000 off prices or ring 094 686 627 (24 hour ansafone).
ACCESS CARD ORDERS ACCEPTED

Personal Computer World
CENTURY

The Spectrum Handbook

DR TIM LANGDELL

A comprehensive and accessible introduction to the ZX Spectrum. A handbook and program library in one – *The Spectrum Handbook* contains more games and application programs than any other book on the market and contains over *80 programs.*

224 pp £4.95 (paperback)
A Century/PCW joint publication

ORDER FORM
To: Department 01C, George Philip Services Ltd, Arndale Road, Wick, Littlehampton, West Sussex BN17 7EN.

Please send me ________ copy/copies of THE SPECTRUM HANDBOOK by TIM LANGDELL at £5.45 per copy (post paid)

I enclose my cheque/postal order for £ ________
(payable to George Philip)

Name ________________________________

Address ________________________________

____

____

Please allow 28 days for delivery

ZX81/ZX SPECTRUM HARDWARE



PROFESSIONAL DESKTOP CONSOLE FOR THE ZX81. IT RAISES AND TILTS THE TV TO MINIMISE EYE-STRAIN, HOLDS IN PLACE THE ZX81 PRINTER, RAM PACKS (AND OTHER ADD-ONS), POWER SUPPLY, TAPE RECORDER AND HIDES MOST OF THE WIRING.

PROVIDES STORAGE SPACE FOR CASSETTES, PENCILS, ETC.

OPTIONAL: TWO BUILT-IN ILLUMINATED POWER SWITCHES. SIMILAR UNIT FOR THE ZX SPECTRUM ALSO AVAILABLE WHICH HOLDS ZX MICRODRIVE IN PLACE.

TV Stand removable for access to add-ons, wiring, etc.
Can be used separately from Base.

Date ....

YOUR NAME AND ADDRESS (in Capital Letters Please) ....

....

....

....

QTY	DESCRIPTION	ITEM PRICE	TOTAL
	Console for ZX81	£29.95	
	Console for ZX Spectrum	£30.95	
	Power Switch for Tape Recorder	£3.00	
	Power Switch for ZX81/ZX Spectrum	£3.00	
		Sub Total	
		Carriage	£3.00
		Total Due	

Cheques/P.O. made payable to:
COMPUTERLOCK, 2 WYCHPERRY ROAD, HAYWARDS HEATH, WEST SUSSEX RH16 1HJ
Telephone: (0444) 451986

All Prices inclusive of V.A.T. Allow 28 days for delivery.

MACHINE SPECIFICATIONS

ZX80

Dimensions
Width 174mm (6.85 in)
Depth 218mm (8.58 in)
Height 38 mm (1. 5 in)
Weight 300g (10.5oz)

Microprocessor/Memory
Z80A 3.25 MHz clock
ROM: 4K bytes containing BASIC
RAM: 1K bytes internal, externally expandable to 16K bytes.

Display
Requires an ordinary domestic black and white colour TV. The lead supplied connects between the ZX80 and your TV's aerial socket. The display organisation is 24 lines of 32 characters per line showing black characters on a white screen. The ZX80 does not connect to a printer.

Programming
Programs can be entered on the keyboard or loaded from cassette. The ZX80 has automatic "wrap round" so lines of program can be any length but not multi-statement lines.

Syntax check
The syntax of the entered line is checked character by character. A syntax error cursor marks the first place the syntax breaks down if there is an error. Once any errors have been edited out the syntax error cursor disappears. Only syntax error-free lines of code are accepted by the ZX80.

Graphics
Total of 22 graphics symbols giving 48 x 64 pixels resolution consisting of 10 symbols plus space and inverses. Includes symbols for drawing bar charts. Under control of your BASIC program any character can be printed in reverse field.

Editing
The line edit allows you to edit any line of program or input including statement numbers. The edit and cursor control keys are EDIT, RUBOUT, HOME.

Arithmetic
Arithmetic operators +,–,x, ÷ exponentiate. Relational operators <, >, = , yielding 0 or – 1. Logical operators AND OR NOT yielding boolean result. Relational operators also apply to strings. ZX80 BASIC uses 16 bit two's complement arithmetic (± 32767).

Variables
Numeric variable names may be any length, must begin with a letter and consist of alphanumerics. Every character in the name is compared thus an infinity of unique names is available.

String variables may be assigned to or from, shortened but not concatenated. String variable names are A$ – Z$. Strings do not require a dimension statement and can be any length.

Arrays have a maximum dimension of 255 (256 elements) each. Array names consist of a single letter A–Z.

Control variable names in FOR. . . NEXT loops consist of a single letter A–Z.

Expression evaluator
The full expression evaluator is called whenever a constant or variable is encountered during program execution. This allows you to use expressions in place of constants especially useful in GOTOs, GOSUBs, FOR. . . NEXT etc.

Immediate mode
The ZX80 will function in the "calculator mode" by immediately executing a statement if it is not preceded with a line number.

Cassette interface
Works with most domestic cassette recorders. The transfer rate is 250 baud using a unique tape-recording format. Other systems are not compatible with the ZX80's. The ZX80 also SAVEs the variables as well as the program on cassette. Therefore you can save the data for updating next time the program is executed. The ZX80 does not support separate data files. The lead supplied with the ZX80 is fitted with 3.5mm jack plugs.

Expansion bus
At the rear has 8 data, 16 address, 13 control lines from the processor and 0v, 5v, 9-11v, Ø and internal memory control line. These signals enable you to interface the ZX80 to your own electronics, PIO, CTC, SIO if you want I/O ports etc.

Power supply
The ZX80 requires approximately 400mA from 7–11v DC. It has its own internal 5v regulator.

TV standard
The ZX80 is designed to work with UHF TVs (channel 36)and is the version required for use in the United Kingdom. The ZX80 USA is designed to work with a VHF TV(American channel 2. European channel 3) and is the version required for the American TV system, also for countries without UHF.

ZX81

Dimensions
Width 167mm (6.32 in)
Depth 175mm (6.80 in)
Height 40 mm (1.57 in)
Weight 350 gms (12.15 oz)

Microprocessor/Memory
Z80A 3.25 MHz clock
ROM: Containing 8K BASIC interpreter
RAM: 1K bytes internal, externally expandable to 16K bytes.

Keyboard
40 key touch-sensitive membrane. Using function mode and single press key-word system, this gives the equivalent of 91 keys and also graphics mode allows an additional 20 graphical and 54 inverse video characters to be entered directly.

Display
Requires an ordinary domestic black and white or colour TV. The aerial lead supplied connects the ZX81 to the TV aerial socket. The display is organised as 24 lines of 32 characters with black characters on a white background.

Two mode speeds
The ZX81 can operate in two software-selectable modes - FAST and NORMAL. FAST is ideal for really high-speed computing. In NORMAL mode however the ZX81 allows continuously moving, flicker-free animated displays

Printer
The 8K ROM will permit instructions (LPRINT, LLIST and COPY) to drive the Sinclair ZX Printer.

Programming
Programs can be entered via the keyboard or loaded from cassette. Programs and data can be saved onto cassette so that they

are not lost when the ZX81 is turned off.

Syntax check
The syntax of a line of program is checked on entry. A syntax error cursor marks the first place the syntax breaks down if there is an error. The syntax error cursor disappears when errors have been corrected. Only lines free from syntax errors will be entered into the program.

Graphics
Apart from the 20 graphics characters, space and its inverse, the display may also be divided into 64 x 44 pixels, each of which may be 'blacked' in or 'whited' out under program control.

Editing
A line editor allows you to edit any line of program or input, including program line numbers. Lines may be deleted, increased or decreased in size.

Arithmetic
Arithmetic operators +, –, x, ÷, exponentiate. Relational operators =, < >, >, <,< =,>=, may compare string and arithmetic variables to yeild 0 (False) or 1(True). Logical operators AND, OR, NOT yield boolean results.

Floating-point numbers
Numbers are stored in 5 bytes in floating-point binary form giving a range of $\pm 3 \times 10^{-39}$ to $\pm 7 \times 10^{38}$ accurate to 9½ decimal digits.

Scientific functions
Natural logs/antilogs; SIN, COS, TAN and their inverses;SQR; e^x.

Variables

Numerical:	any letter followed by alphanumerics
String:	A$ to Z$
FOR-NEXT loops:	A–Z (loops may be nested to any depth.
Numerical arrays:	A–Z
String arrays:	A$ to Z$

Arrays
Arrays may be multi-dimensional with subscripts starting at 1.

Expression evaluator
The full expression evaluator is called whenever an expression, constant or variable is encountered during program execution. This powerful feature allows use of expressions in place of constants and is especially useful in GOTO, GOSUB etc.

Command mode
The ZX81 will execute statements immediately, enabling it to perform like a calculator.

Cassette interface
Works using domestic cassette recorders. The transfer rate is 250 baud and uses a unique recording format not compatible with other systems. The ZX81 will save the data as well as the program to avoid the need to re-enter the data when the program is next loaded.,
ZX81 will search through a tape for the required program). The cassette leads supplied have 3.5 mm jack plugs.

Expansion port
At the rear, this has the full data, address and control buses from the Z80A CPU as well as OV, +5V, +9V,Ø̄ and the memory select lines. These signals enable you to interface the ZX81 to the Sinclair 16K RAM pack and ZX printer.

Power supply
The ZX81 requires approximately 420mA at 7–11V DC. It has its own internal 5V regulator. The ready assembled ZX81 comes complete with a power supply. The ZX81 kit does not include a power supply.

TV standard
The ZX81 is designed to work with UHF TVs (channel 36) 625 lines.

ZX SPECTRUM

Dimensions

Width 233 mm
Depth 144 mm
Height 30 mm

CPU/Memory

Z80A microprocessor running at 3.5 MHz. 16K-byte ROM containing BASIC interpreter and operating system.
16K-byte RAM (plus optional 32K-byte RAM on internal expansion board) or 48K-byte RAM.

Keyboard

40-key keyboard with upper and lower case with capitals lock feature. All BASIC words obtained by single keys, plus 16 graphics characters, 22 colour control codes and 21 user-definable graphics characters. All keys have auto repeat.

Display

Memory-mapped display of 256 pixels x 192 pixels; plus one attributes byte per character square, defining one of eight foreground colours, one of eight background colours, normal or extra brightness and flashing or steady. Screen border colour also settable to one of eight colours. Will drive a PAL UHF colour TV set, or black and white set (which will give a scale of grey), on channel 36.

Sound

Internal loudspeaker can be operated over more than 10 octaves (actually 130 semitones) via basic BEEP command. Jack sockets at the rear of computer allow connections to external amplifier/ speaker.

Graphics

Point, line, circle and arc drawing commands in high-resolution graphics.
16 pre-defined graphics characters plus 21 user-definable graphics characters. Also functions to yield character at a given position, attribute at a given position (colours, brightness and flash) and whether a given pixel is set. Text may be written on the screen on 24 lines of 32 characters. Text and graphics may be freely mixed.

Colours

Foreground and background colours, brightness and flashing are set by BASIC INK, PAPER, BRIGHT and FLASH commands. OVER may also be set, which performs an exclusive – or operation to overwrite any printing or plotting that is already on the screen. INVERSE will give inverse video printing. These six commands may be set globally to cover all further PRINT, PLOT, DRAW or CIRCLE commands, or locally within these commands to cover only the results of that command. They may also be set locally to cover tex printed by an INPUT statement. Colour-control codes, which m be accessed from the keyboard, may be inserted into text or program listing, and when displayed will override the globally set colours until another control code is encountered. Brightness and flashing codes may be inserted into program or text, similarly. Colour-control codes in a program listing have no effect on its execution. Border colour is set by a BORDER command. The eight colours available are black, blue, red, magneta, green, cyan, yellow and white. All eight colours may be present on the screen at once, with some areas flashing and others steady, and any area may be highlighted extra bright.

Screen

The screen is divided into two sections. The top section – normally the first 22 lines – displays the program listing or the results of program or command execution. The bottom section – normally the last 2 lines – shows the command or program line currently being entered, or the program line currently being edited. It also shows the report messages. Full editing facilities of cursor left, cursor right, insert and delete (with auto-repeat facility) are available over this line. The bottom section will expand to accept a current line of up to 22 lines.



Mathematical Operations And Functions

Arithmetic operations of +, −, ×, ÷, and raise to a power. Mathematical functions of sine, cosine, tangent and their inverses; natural logs and exponentials; sign function, absolute value function, and integer function; square root function, random number generation, and pi.
Numbers are stored as five bytes of floating point binary — giving a range of $+3 \times 10^{-39}$ to $+7 \times 10^{38}$ accurate to 9½ decimal digits. Binary numbers may be entered directly with the BIN function. =, >, <, >=, <= and <> may be used to compare string or arithmetic values or variables to yield 0 (false) or 1 (true). Logical operators AND, OR and NOT yield boolean results but will accept 0 (false) and any number (true).
User-definable functions are defined using DEF FN, and called using FN. They may take up to 26 numeric and 26 string arguments, and may yield string or numeric results.
There is a full DATA mechanism, using the commands READ, DATA and RESTORE.
A real-time clock is obtainable.

String Operations And Functions

Strings can be concatenated with +. String variables or values may be compared with =, >, <, >=, <=, <> to give boolean results. String functions are VAL, VAL$, STR$ and LEN. CHR$ and CODE convert numbers to characters and vice versa, using the ASCII code. A string slicing mechanism exists, using the form a$ (x TO y).

Variable Names

Numeric — any string starting with a letter (upper and lower case are not distinguished between, and spaces are ignored).
String — A$ to Z$.
FOR-NEXT loops — A-Z.
Numeric arrays — A-Z.
String arrays — A$ to Z$.
Simple variables and arrays with the same name are allowed and distinguished between.

Arrays

Arrays may be multi-dimensional, with subscripts starting at 1. String arrays, technically character arrays, may have their last subscript omitted, yielding a string.

Expression Evaluator

A full expression evaluator is called during program execution whenever an expression, constant or variable is encountered. This allows the use of expressions as arguments to GOTO, GOSUB, etc.
It also operates on commands allowing the ZX Spectrum to operate as a calculator.

Cassette Interface

A tone leader is recorded before the information to overcome the automatic recording level fluctuations of some tape recorders, and a Schmitt trigger is used to remove noise on playback.
All saved information is started with a header containing information as to its type, title, length and address information. Program, screens, blocks of memory, string and character arrays may all be saved separately.
Programs, blocks of memory and arrays may be verified after saving.
Programs and arrays may be merged from tape to combine them with the existing contents of memory. Where two line numbers or variables names coincide, the old one is overwritten.
Programs may be saved with a line number, where execution will start immediately on loading.
The cassette interface runs at 1500 baud, through two 3.5 mm jack plugs.

Expansion Port

This has the full data, address and control busses from the Z80A, and is used to interface to the ZX Printer, the RS232 and NET interfaces and the ZX Microdrives. IN and OUT commands give the I/O port equivalents of PEEK and POKE.

ZX81 Compatibility

ZX81 BASIC is essentially a subset of ZX Spectrum BASIC. The differences are as follows.
FAST and SLOW: the ZX Spectrum operates at the speed of the ZX81 in FAST mode with the steady display of SLOW mode, and does not include these commands.
SCROLL: the ZX Spectrum scrolls automatically, asking the operator "scroll?" every time a screen is filled.
UNPLOT: the ZX Spectrum can unplot a pixel using PLOT OVER, and thus achieves unplot.
Character set: the ZX Spectrum uses the ASCII character set, as opposed to the ZX81 non-standard set.

• READERS •

Reach over 70,000 **ZX** Enthusiasts.
Swop, sell or buy your **ZX** Hardware, Software, Peripherals etc. in our
NEW CLASSIFIED SECTION

STARTING SOON IN

ZX COMPUTING

Britain's Biggest Magazine for the Sinclair User.

PHONE
BRIDGETTE ON

01-437 1002

FOR DETAILS

ADVERTISEMENT INDEX

Push your Sinclair to the limit

ZX81/SPECTRUM ARCADE GAMES

NEW **ZX INVASION FORCE** (ZX81)
Use your skill to fire through the energy band to destroy the menacing alien ship — **£3.95**

NEW **SPEC GALAXIANS**
SPEC INVADERS
SPEC GOBBLEMAN
These three exciting high-speed classic games incorporate hi-resolutiongraphics and sound to bring you the best in arcade action at only **£4.95** each (16/48K)

NEW **SPEC FROG 5/SHOWDOWN** (16/48K)
Hop across the dangerous road avoiding the lethal cars, then lorries. Go back to the wild west for a gunfight battle amongst cacti and wagons. Features include western music. **£4.95**

NEW **NAMTIR RAIDERS** (ZX81)
High speed, quick action arcade game with four separate groups of attackers. **£3.95**

Galaxians and Gobbleman also available for ZX81 at **£3.95** each.

UTILITIES

NEW **TOOLKIT**
Nine powerful new functions for your ZX81 (including renumber lines (goto, gosub), program merge, string finding and replacement. **£5.95**

SPEC BUG
A 30 in 1 machine code tool and disassembler for your ZX Spectrum. **£6.95**

ZX BUG
A 30 in 1 machine code tool and disassembler for your ZX81. **£6.95**

Improve your machine code programs with this new 48K Spectrum Assembler

Just released by Artic, this new professional quality assembler is also available for 16K ZX81 and features:—

• Word-processor-like text editor. • High-speed, versatile two-pass mnemonic assembler with labels and detailed error-trapping. Will assemble to any address. • Essential support facilities such as memory-edit, register inspect/modify and search for de-bugging. • Full output to ZX-printer. • Memory-status report and comprehensive user-manual.

ONLY **£9.95**

ADVENTURE GAMES

JUST RELEASED for 48K SPECTRUM
Still available for ZX81

INCA CURSE (Adventure B)
In a jungle clearing you come across an ancient Inca temple. Your mission to enter, collect the treasure and escape alive. But beware! Includes a cassette save routine. **£6.00**

SHIP OF DOOM (Adventure C)
You are unavoidably drawn to an alien cruiser. Can you reach the control room and free yourself? Or will they get you first? Includes a cassette save routine. **£7.00**

ESPIONAGE ISLAND (Adventure D)
While on a reconnaisance mission your plane loses control and you are forced to land. Can you survive and escape with the island's hidden secret? **£8.00**

PLANET OF DEATH (Adventure A)
You find yourself stranded on an inhabited alien planet. Can you reach your ship and escape? **£5.00**

ZX FORTH

REDUCED TO £29.95

Supplied on cassette with editor cassette, user manuals and keyboard overlay. ZX Forth combines the simplicity of basic with the speed of machine code

THE ULTIMATE ADVENTURE FOR THE SPECTRUM/ZX81 for only £9.95

KRAKIT™

AND THE CHANCE TO WIN £10,000 or more IF YOU

KRAKIT TRADE MARK OF INT PUBLISHING & SOFTWARE INC

RULES AVAILABLE FROM ARTIC

Your eccentric father has left you £10,000 in his will. In order to claim your windfall you must solve twelve clues and gain access to a bank account in which the money has actually been deposited. Be the first to crack the puzzle and the prize is yours! Plus you will win two tickets to the city of the secret KRAKIT™ vault location. The prize money is increased weekly. A telephone number will be supplied so you have the opportunity to find out just how much you could win.

DON'T MISS YOUR CHANCE TO WIN A FORTUNE!

A GREAT CHESS GAME FOR YOUR ZX81 OR SPECTRUM*



ZX CHESS I
Very popular machine code program, with six levels of play and an analysis option. Available for ZX81. **£6.50**

ZX CHESS II
A new improved version, with a faster response time, seven levels of play, analysis option and in addition a recommended move option. **£9.95**

ZX 1K CHESS
An incredible game in 1K for only **£2.95**

SPECTRUM CHESS now only **£9.45**

NEW **SPECTRUM VOICE CHESS**

This incredible program talks you through your game. It tells you its move, recommends a move, and contains a range of facetious comments. The game is based on the highly successful ZX Chess II and is offered at the Christmas offer price of **£9.95**.

*SPECTRUM CHESS 48K

ALWAYS AHEAD WITH ZX81/SPECTRUM SOFTWARE

Cheques & P.O. made payable to: Artic Computing Limited. Dealer enquiries welcome.
Please state whether your order is for ZX81 or Spectrum. Send S.A.E. (6in × 9in) for catalogue.

VISA

ARTIC COMPUTING LTD.
396 JAMES RECKITT AVENUE, HULL, N. HUMBERSIDE,

To: Artic Computing Ltd., 396 James Reckitt Avenue, Hull, N. Humberside. HU8 0JA

Please supply ....

Cheque for total amount enclosed ....

Name ....

Address ....

ZX12